1992年11月1日発行（毎月1回1日発行）第11巻11号通巻127号　昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

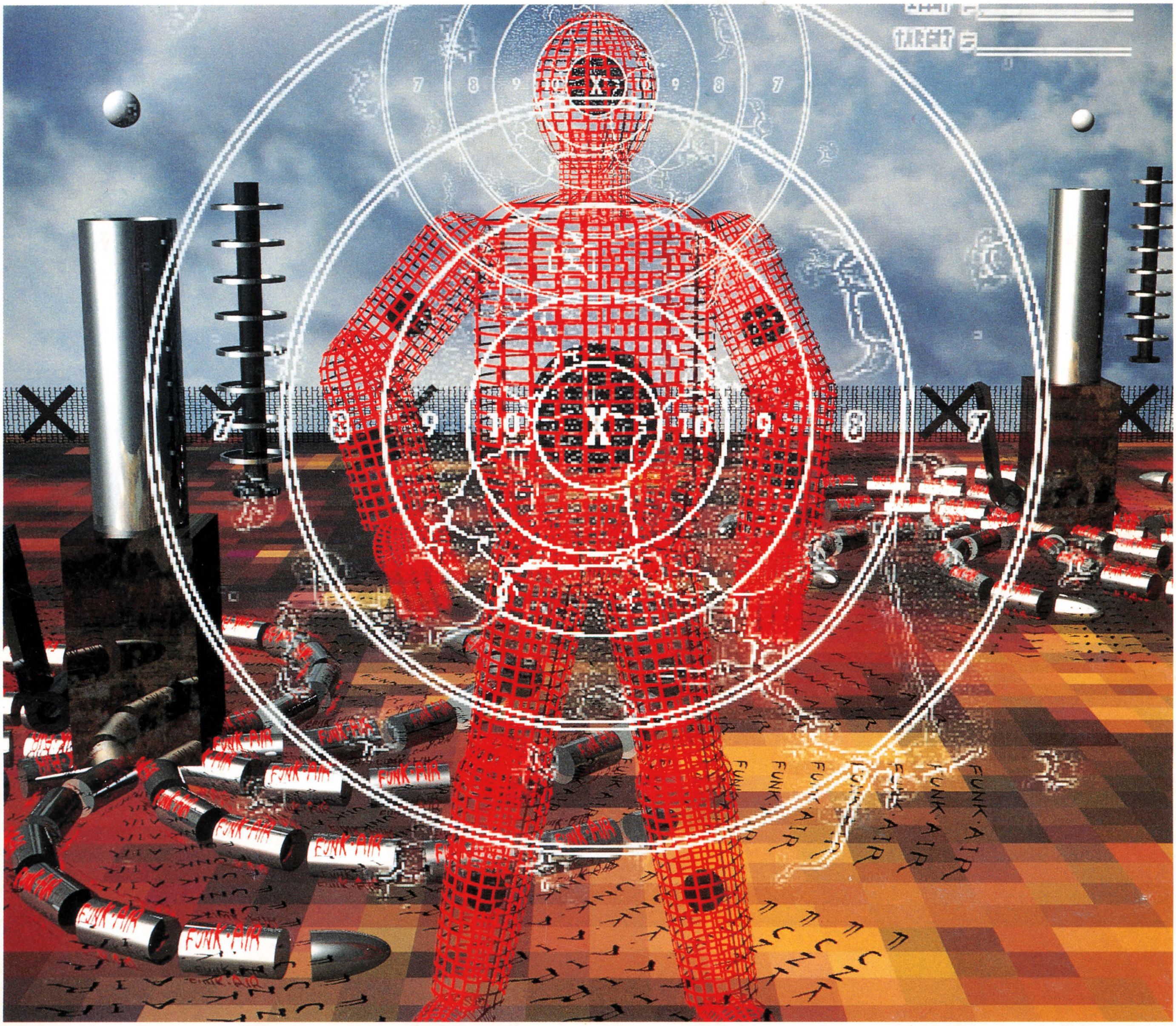
Oh!X

特集 ゲームマネージメント

X68000用BEMS/ゲーム画面保存プログラム SAVESC.SYS
CARDDRV対応カードゲーム サーティーン・ダウン
新製品紹介 CHART PRO-68K/データショウレポート

11
1992

SOFT BANK オー！エックス
定価600円



SHARP

目の付けどころが、
シャープでしょ。

“感性”咲かせるワー

POWER WORKSTATION

インテリジェントなパフォーマンスを誇るX68000Compact XVIと多彩にラインアップされたペリフェラル。感性を刺激するクリエイティブなワークステーション環境が自在に構築できます。

- パーソナルワークステーション(2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)
  CZ-674C-H(グレー) 標準価格**298,000**円(税別)
- 15型カラーディスプレイテレビ
  CZ-614D-TN(チタンブラック)・**-BK**(ブラック) 標準価格**135,000**円(税別)
  ■ディスプレイテレビ/CZ-6TU用RGBケーブル**CZ-6CR1** 標準価格**4,500**円(税別)
  ■ディスプレイテレビ/CZ-6TU用TVコントロールケーブル**CZ-6CT1** 標準価格**5,500**円(税別)
- 80MB内蔵用ハードディスクドライブ
  CZ-68HA 好評発売中
- 5.25インチ増設用フロッピーディスクドライブ
  CZ-6FD5 標準価格**99,800**円(税別・接続ケーブル同梱)
- 光磁気ディスクユニット
  CZ-6MO1 標準価格**450,000**円(税別)
  ■SCSI変換ケーブル**CZ-6CS1** 標準価格**12,000**円(税別)
- 2MB増設RAMボード
  CZ-6BE2D 標準価格**54,800**円(税別・取り付け費別)
  ■2MB増設RAM**CZ-6BE2B** 標準価格**54,800**円(税別・取り付け費別)×2
  ■数値演算プロセッサ**CZ-6BP2** 標準価格**45,800**円(税別・取り付け費別)
- 48ドット熱転写カラー漢字プリンタ
  CZ-8PC5-BK(ブラック) 標準価格**96,800**円(税別)
- MIDIボード
  CZ-6BM1A 標準価格**26,800**円(税別)
- インテリジェントコントローラ
  CZ-8NJ2 標準価格**23,800**円(税別)

X68000 見・体・験フェア [コンパクト祭り]

好奇心旺盛な浪速人のクリエイティブ・マインドを刺激するビッグイベント、X68000見体験フェア「コンパクト祭り」がいよいよ、大阪で行なわれます。パソコンならおまかせのキミ、そしてパソコン未体験のひとも――――Let's all join in.

●「第1回全日本X68000芸術祭」作品紹介など。

フェア当日X68000お買い上げの方にすてきなプレゼントを用意しております。

資料請求券
X68000ペリフェラル
Oh!X
11係

クステーション環境。

X68000
PERSONAL WORKSTATION・XVI
Compact

GRAPHIC WORKSTATION

- パーソナルワークステーション(2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)
  CZ-674C-H(グレー) 標準価格**298,000**円(税別)
- 21型カラーディスプレイ **CU-21HD** 標準価格**148,000**円(税別)
- 80MB内蔵用ハードディスクドライブ **CZ-68HA** 好評発売中
- 光磁気ディスクユニット **CZ-6MO1** 標準価格**450,000**円(税別)
  ■SCSI変換ケーブル**CZ-6CS1** 標準価格**12,000**円(税別)
- 2MB増設RAMボード **CZ-6BE2D** 標準価格**54,800**円(税別・取り付け費別)
  ■2MB増設RAM**CZ-6BE2B** 標準価格**54,800**円(税別・取り付け費別)×2
  ■数値演算プロセッサ**CZ-6BP2** 標準価格**45,800**円(税別・取り付け費別)
- カラーイメージスキャナ
  CZ-8NS1 標準価格**188,000**円(税別)
  ■スキャナ用パラレルボード**CZ-6BN1** 標準価格**29,800**円(税別)
- カラーイメージジェット
  IO-735X-B(ブラック) 標準価格**248,000**円(税別)
  ■接続ケーブル**IO-73CX** 標準価格**5,500**円(税別)

STANDARD WORKSTATION

- パーソナルワークステーション
  (2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)**CZ-674C-H**(グレー) 標準価格**298,000**円(税別)
- 14型カラーディスプレイ**CZ-608D-H**(グレー) 標準価格**94,800**円(税別)
- 5.25インチ増設用フロッピーディスクドライブ**CZ-6FD5** 標準価格**99,800**円(税別・接続ケーブル同梱)

TFT COLOR LCD WORKSTATION

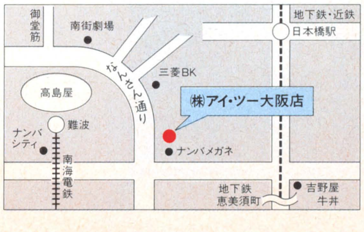
- パーソナルワークステーション
  (2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)**CZ-674C-H**(グレー) 標準価格**298,000**円(税別)
- 10.4型カラー液晶ディスプレイ**LC-10C1-H**(グレー) 標準価格**598,000**円(税別)
  ■接続ケーブル**AN-1515X** 標準価格**4,200**円(税別)

※カラー液晶ディスプレイを接続してご使用の場合、SX-WINDOW上のアプリケーション利用に限定されます。

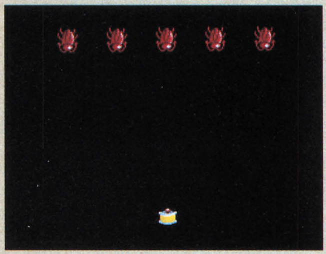
開催日時：**10月24日(土)・25日(日)** 11:00〜19:00

会　　場：**㈱アイ・ツー大阪店**（南海線なんば駅徒歩1分）
大阪市中央区難波千日前15-18 ☎06-632-0012㈹

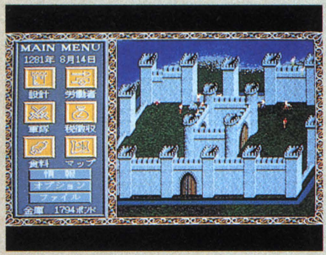
■主催/アイ・ツー■お問い合わせ/シャープエレクトロニクス販売㈱ 近畿統轄営業部パソコンシステム営業部 ☎06-631-1181㈹ 担当・金光



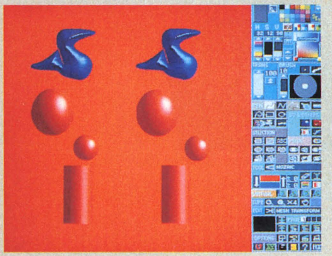
■お問い合わせは…電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06) 621-1221(大代表)
電子機器事業本部AVCシステム事業推進室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表) シャープ株式会社



特集　ゲームマネージメント

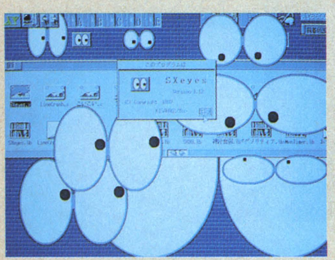


キャッスルズ

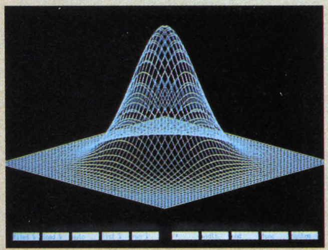


大人のための X 68000

カードゲーム



THE USER'S WORKS



(て)のショートプロばーてぃ

Oh!X

C O N T

●特集

73 ゲームマネージメント

74 概論
ゲームシステムの構成とは　中野修一

76 よりよいゲームを作るために
違いのわかるアソビロジー　伊谿見あきら

78 リアルタイムゲームのための
“基本”スプライトルーチン作成法　横内威至

85 ゲームマネージャの作成
BEMSによるキャラクタ管理　影山裕昭

93 考え方と実例を探って
オブジェクト指向を取り入れる　丹　明彦

97 シナリオ管理の手法
アドベンチャーシステムを考える　香坂正嗣

101 プログラム速度制御関数
XVI.FNC　紙山　満

103 BASICのスプライト処理高速化
SPRITE2.FNC　中村文洋

107 グラフィック画面の3D回転
3DRT_256　渡辺貴生

●カラー紹介

57 SHOW REPORT
データショウ'92

58 新製品紹介
CHART PRO-68K　金子俊一

62 THE USER'S WORKS
SX福袋

63 Oh!X Graphic Gallery
DōGA・CGアニメーション講座

●読みもの

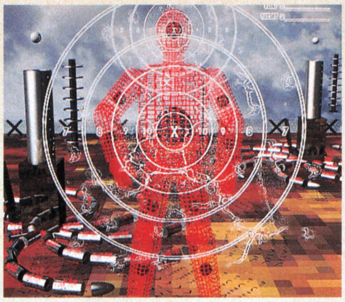
144 X-OVER NIGHT　第28話
バックアップ　高原秀己

152 第64回　知能機械概論―お茶目な計算機たち―
アルゴリズムとしての進化論　有田隆也

154 猫とコンピュータ　第74回
猫に偏差値　高沢恭子

〈スタッフ〉

●編集長／前田　徹　●副編集長／植木章夫　●編集／浅井研二　山田純二　豊浦史子　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　吉田賢司　影山裕昭　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　宮島　靖　金子俊一　浦川博之　石上達也　柴田　淳　御木徳高　瀧　康史　●カメラ／杉山和美　●イラスト／山田晴久　寺尾響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　ADGREEN　●校正／グループごじら



表紙絵：塚田　哲也

E　N　T　S

●THE SOFTOUCH

14 SOFTWARE INFORMATION
新作ソフトウェア/TOP10

16 TREND ANALYSIS

GAME REVIEW
18 キャッスルズ　浦川博之
20 シュートレンジ　高橋哲史
22 ポピュラスII　中野修一
25 サンダーレスキュー　八重垣那智

26 AFTER REVIEW
シムアース

●シリーズ全機種共通システム

45 THE SENTINEL

46 実践Small-C講座(7) EDIT　石上達也

●連載/紹介/講座/プログラム

28 大人のためのX68000［第25回］
いよいよ本題，立体視　荻窪　圭

32 ゲーム画面保存プログラム
SAVESC.SYS　杉本利貴

60 響子 in CG わ～るど［第18回］
風の吹いた日　寺尾響子

64 DōGA CGアニメーション講座 ver. 2.50（第3回）
打倒TORNADOへの第一歩（後編）　かまたゆたか

115 X68000用CARDDRV対応カードゲーム
サーティーン・ダウン　大久保明広

119 よいこのSX-WINDOW講座（第11回）
タスク間通信をためす　中森　章

125 ハードウェア工作入門〈29〉コンピュータアーキテクチャ編
外部電源の作成　三沢和彦

129 Creative Computer Music入門（14）
分厚いハーモニーを作る　瀧　康史

Oh!X LIVE in '92
134 BALROGのテーマ（X68000・Z-MUSIC用 SC-55対応）　渡辺一彦
ARCADIA（X68000・Z-MUSIC+PCM8用）　岡田一彦
スーパーマリオ 地上のテーマ（X1・MusicBASIC用）　中村直哉

142 もうひとつの情報源
海外雑誌の読み方　秋川　涼

146 (で)のショートプロぱーてぃ　その38
文明の窓・テレビであるのだ　古村　聡

150 ANOTHER CG WORLD　寺尾響子

ペンギン情報コーナー……156
FILES Oh!X……158
Oh!X質問箱……160
STUDIO X……162
編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey……166

1992 NOV.
11

UNIXはAT & T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mplus, CP/M-86 CP/M-68K, CP/M-8000, DR-DOSはデジタルリサーチ
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACRO80, MS C, MS-WindowsはMICROSOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事会
TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTERNATIONAL
LSI CはLSI JAPAN
HuBASICはハドソンソフト
の商標です。その他，プログラム名,CPUは一般に各メーカーの登録商標です。本文中では"TM","R"マークは明記していません。
本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権上，PDSと明記されたもの以外，個人で使用するほかの無断複製は禁じられています。

■広告目次

アイビット電子……175(上)
アクセス……176
計測技研……171
J & P……表3
シャープ……表2・表4・1・4・5
九十九電機……11
パソコンプラザオクト……172・173
ビクター音楽産業……7
P & A……8・9
ブラザー工業……6
マイコンショップ川口……174
満開製作所……170
ラインシステム……175(下)

SHARP

目の付けどころが、
シャープでしょ。

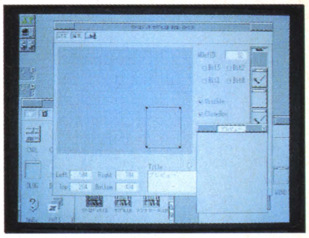
待望のSX-WINDOW
開発支援ツール、登場。

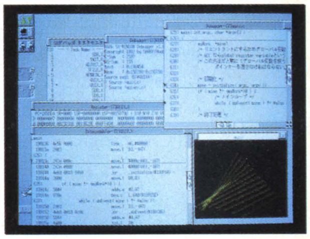
SX-WINDOW 開発キット Workroom SX-68K

NEW

CZ-288LWD 10月発売予定

SX-WINDOW用のソフト開発に必要な開発ツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解ができる33種のサンプルプログラム付き。また各マネージャ解説と関数リファレンスの詳細なマニュアルも装備しています。





※本ソフトのご使用に際しては、メインメモリ4MB以上、SX-WINDOW ver2.0以上、C compiler PRO-68K ver 2.1以上が必要です。

キット構成

■開発ツール

●SXデバッガ
SX-WINDOW上で複数のプログラムを同時にデバッグすることができるソースコードデバッガ。

●リソースエディタ
SX-WINDOW上のリソースをリソースタイプごとの編集ウィンドウでビジュアルに作成・編集が可能。

●リソースリンカ
Cコンパイラやアセンブラで作成したリソースデータファイル(オブジェクトファイル)をリンクしてリソースファイルを作成。

●サンプルメイク
サンプルプログラムのコンパイル作業をSX-WINDOW上から、XC ver2のMAKE. Xを呼び出して、自動実行する簡易メイクユーティリティ。

■サンプルプログラム

●基礎編(23種)
各マネージャの基本的な機能のみを用いた基本動作の理解。

●応用編(4種)
基礎編での基本機能を応用した簡単なアプリケーションの作成。

●実用編(6種)
基礎/応用編での機能を駆使した、実用的なアプリケーションの作成。

■その他のファイル

●インクルードファイル
Cコンパイラとアセンブラ用の関数定義、データ定義ファイル。

●ライブラリファイル
Cコンパイラ用の関数ライブラリ。

マニュアル

●ユーザーズマニュアル●プログラマーズマニュアル●ファンクションリファレンス●ライブラリリファレンス

X68000
PERSONAL WORKSTATION・XVI
Compact

本体+キーボード+マウス
2HD3.5インチFDDタイプ
CZ-674C-H(グレー) 標準価格298,000円(税別)

14型カラーディスプレイ(ドットピッチ0.28mm)
CZ-608D-H(グレー) 標準価格94,800円(税別)

●5.25インチ増設用
フロッピーディスクドライブ
CZ-6FD5
標準価格99,800円・税別
〔接続ケーブル同梱〕

●ディスプレイテレビ/CZ-6TU用RGBケーブル
CZ-6CR1 標準価格4,500円・税別

●ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル
CZ-6CT1 標準価格5,500円・税別

●SCSI変換ケーブル CZ-6CS1 標準価格12,000円・税別

資料請求券
CZソフト
OH!X
11係

開いてくださいウィンドウ、触れてくださいインテリジェンス。
さらに拡がる、SXワールド。

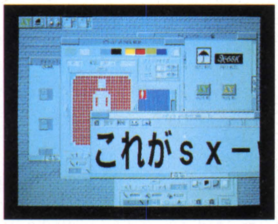
●アウトラインフォント対応、ひらかれたウィンドウ環境。

SX-WINDOW ver2.0

CZ-287SS 標準価格12,800円(税別)

フォントマネージャを装備して待望のアウトラインフォントに対応。画面スクロール機能により、表示画面よりワイドなデスクトップ空間を駆使。アプリケーションのハンドリングに便利なシンボルトレイやアイコンメンテ、パターンエディタなど便利機能満載。

※SX-WINDOW ver1.0(CZ-259SS)およびSX-WINDOW ver1.1(CZ-278SS)をお持ちの方には有償バージョンアップを行います。



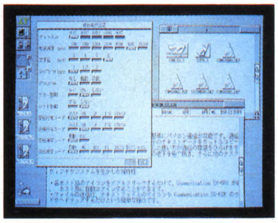
●マルチタスク機能をはじめ、通信環境がさらに充実。

Communication SX-68K

NEW

CZ-272CWD 標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションソフトを実行中でも簡単に通信が可能。また、ホスト局をクリックするだけの自動ログイン機能、初心者にも簡単なプログラム機能、最新モデム(20種類)もフルサポートしています。



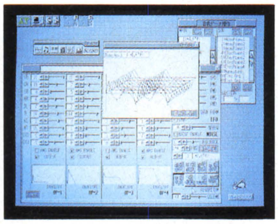
●多彩なサウンドクリエイトを実現するFM音源サウンドエディタ。

SOUND SX-68K

NEW

CZ-275MWD 標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成・変更ができるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。まさにミキサー感覚で音創りが楽しめるツールです。



●ウィンドウ対応グラフィックツール。

Easypaint SX-68K

CZ-263GWD 標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。

※SX-WINDOW対応ソフトの動作には、メインメモリ2MBおよびSX-WINDOW ver1.1以上が必要です。

充実のPROシリーズ

●ビジネスグラフチャート

CHART PRO-68K

NEW

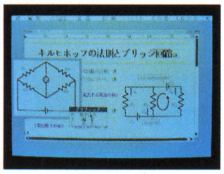
CZ-267BSD 標準価格38,000円(税別)

各種データベースで作成したデータをもとに、多彩なグラフが作成できます。3次元表示やグラフの複合機能も装備。データはMultiword,Press Conductor PRO-68Kに取り込むこともできます。

●グラフィック機能搭載の本格派ワープロ

Multiword ver1.1

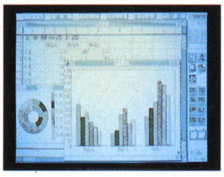
CZ-225BSD 標準価格32,000円(税別)



●簡単操作の統合型表計算ソフト

BUSINESS PRO-68K Popular

CZ-286BSD 標準価格28,000円(税別)



●各種ドライバ、ライブラリを追加

COMPILER ver2.1 PRO-68K

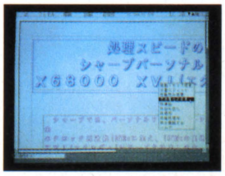
CZ-285LSD 標準価格44,800円(税別)

※有償バージョンアップ対応中。

●各種エディタ装備のレイアウトソフト

PressConductor PRO-68K

CZ-266BSD 標準価格28,000円(税別)



※以上のPROシリーズのソフトの動作にはメインメモリ2MB必要です。

※発売予定のソフトの画面写真は実物とは異なる場合があります。

●お問い合わせは…シャープ㈱電子機器事業本部AVCシステム事業推進室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ。 シャープ株式会社

brother
パソコンソフト自販機
TAKERU
サンダーレスキュー
全5面のステージ！横スクロール・シューティングアクション!!
3重スクロールの美しいグラフィックと巨大ボスキャラ達!!
スムーズ&パワフルな動き！多彩な敵の攻撃パターンを迎えうて!!
難易度高め。何度でもトライできる、ハイコストパフォーマンス！
ステージは、マルチチョイス、得意なボスから攻略せよ!!
MEGATON
10/20発売
X68000版
TAKERU価格 ￥4,800(税込)
■企画/開発：ギミックハウス
チェイスH.Q.
CHASE H.Q.
好評発売中
TAKERU価格
￥7,800(税込)
©TAITO
[Xak II]
Rising of The Redmoon
©1989/1990 MICRO CABIN
TAKERU価格
￥6,800(税込)
©1989/1990
マイクロキャビン
発売中
ブラザー工業株式会社 〒467 名古屋市瑞穂区苗代町2番1号
TAKERU事務局
(052)824-2493
東京営業所(03)5203-7133
大阪営業所(06)252-4234
通信販売
通信販売をご希望の方は、ソフト名・機種名・住所・氏名・電話番号を明記の上TAKERU事務局まで現金書留でお申し込み下さい
代金引換は一度、現金書留で申し込んで頂いた方に案内させて頂きます

Victor
創造の美学と崩落の悲哀……城に託す壮大なロマンの誕生
CASTLES
全米で大ベストセラーを記録した築城シミュレーション!
キャッスルズ
10/23 発売予定
X68000シリーズ マウス対応
FM-TOWNS ㊊2MB RAM
好評発売中
PC-9801VX/UX以降 ㊊バスマウス
各機種 ¥9,800 (税別)
CASTLES PUBLISHED UNDER LICENSE FROM INTERPLAY PRODUCTIONS, INC
© 1991, 1992 INTERPLAY PRODUCTIONS, INC ALL RIGHTS RESERVED © 1992 VICTOR MUSICAL INDUSTRIES, INC
全米で大ヒットを記録した、13世紀のイギリスを舞台に展開する築城シミュレーション・ゲームいよいよX68000にて登場! プレイヤーは、架空の王国アルビオンの城主になって先住民族ケルト人の地に拠点となる城を構築して領土を拡大します。しかし、その途中で様々な問題が発生します。封建制度のもとで強大な力を持つ教会や貴族、さらにはたびたび攻撃をかけてくるケルト人、刻々変化する近隣諸国情勢。それらを解決しながら立派な城を作るために、設計し、職人を集め、給料を払い、さらに住民の税金の取り立てなどを行います。様々な要素を解決しながら物語は進みます。はたして貴方はアルビオンの誉れ高い王として後世に名を残すことが出来るでしょうか?
MAIN MENU
バトルテック
奪われた聖杯
奪われた聖杯を取り戻せ!
広大な宇宙空間に展開されるロボット・ウォーズ!
■アメリカで爆発的なロボットブームを巻き起こした話題作いよいよ登場!
■3Dポリゴンの採用による迫真のバトル・アクション
■プレイヤーが自らコックピットに乗り込みロボットを操縦、リアルなロボット・シミュレーションを体験
■共に戦うクルーとして41人の傭兵から最大3人までの傭兵の採用により、戦略性もゲームの重要な要素
■情報収集と賞金稼ぎによってグレイドの高いメックを手にいれて、150の惑星を舞台に任務を遂行
好評発売中!
X68000対応
(5FD, 3.5FD)
¥9,800 (税別)
© 1992 Activision. All rights reserved. Battletech is registered trademarks of FASA Corporation.
© 1992 Victor Musical Industries, Inc.
Rising Sun
ライジング・サン
好評発売中!
X68000対応
(5FD)
¥9,800 (税別)
野望が渦巻く!! 貴族の世界か武士の時代か…
源氏と平家の壮絶な戦い
■米国生まれのそして米国で大評判を呼んだリアルタイム・シミュレーションゲームがオリジナルよりさらにバージョンアップして登場
■「源 頼朝」か「源 義経」になって平家と戦いながら日本を手中におさめろ!
■変化に富んだ「馬追い」「忍者戦」などの4つのアクション・ゲームとリアルタイムの「合戦」がゲームを盛り上げる
■刻々変化する情勢に的確に反応して軍勢を操りながら平家を倒すか全国の城を制圧
© 1992 VICTOR MUSICAL INDUSTRIES, INC.
発売元: ビクター音楽産業株式会社
通信販売 当社の商品をお近くのパソコンショップでお買い求めになれない場合、商品名、機種名、住所、氏名、電話番号を明記のうえ、下記住所まで定価プラス3%消費税分を現金書留にてお申し込み下さい (送料無料) 〒151 東京都渋谷区千駄ヶ谷2-8-16 ビクター音楽産業㈱〈通信販売係〉

P&Aならではの新品パソコン 5年保証

《業界No.1の"P&Aメンテナンスサポート》

最高の保証システム

①業界最長の新品パソコン5年保証
（※モニター・プリンター３年間保証.// ※一部商品は除きます。）
②中古パソコンの１年間保証
（モニター・プリンター６ヶ月間保証）
③初期不良交換期間3ヶ月
（※新品商品に限らせていただきます）
④永久買取保証
⑤配達の指定OK//
⑥夜間配送もOK//
（※PM6:00～PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

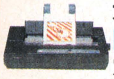
①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利
③月々の支払いは¥1,000より
④9ヶ月先からのスキップ払いOK//
⑤84回までの分割、ボーナス併用OK//
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK//
⑨現金一括払いOK//
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

またまた

《増設メモリー&数値演算プロセッサ》計測技研

1 PRKII-02(2M)……定価¥ 55,000▶特価¥ 39,800
2 PRKII-04(4M)……定価¥ 90,000▶特価¥ 67,000
3 PRKII-06(6M)……定価¥125,000▶特価¥ 92,500
4 PRKII-08(8M)……定価¥160,000▶特価¥119,000
5 PRKII-12(2M)……定価¥ 85,000▶特価¥ 63,000
6 PRKII-14(4M)……定価¥120,000▶特価¥ 89,500
7 PRKII-16(6M)……定価¥155,000▶特価¥114,500
8 PRKII-18(8M)……定価¥190,000▶特価¥141,000
9 MC-68881RC……定価¥ 38,000▶特価¥ 27,000

10/18～11/17

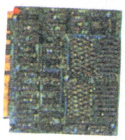


カラーイメージジェット
■IO-735X-B
定価¥248,000
特価¥152,000
（送料・消費税込み¥157,590）

■Z's STAFF PRO 68K Ver.3.0
（ツアイト）（定価¥58,000）
特価 ¥37,500
（送料・消費税込み¥39,140）

■SX-68MII
MIDI
（システムサコム）
定価¥19,800
特価¥13,500
（送料・消費税込み¥14,420）

X68000メモリボード



①SH-6BE1-1M(600C専用)(I/Oデータ)定価¥25,000
（送料・消費税込み¥18,952）……特価¥17,900
②1MB増設RAMボード(ACE/PRO/PROII用)定価¥25,000
（送料・消費税込み¥16,583）……特価¥15,600
③2MB増設RAMボード(拡張スロット用)定価¥50,000
（送料・消費税込み¥32,239）……特価¥30,800
④4MB増設RAMボード(拡張スロット用)定価¥88,000
（送料・消費税込み¥55,620）……特価¥53,500

FDD(5インチ×2基)
■CZ-6FD5
（シャープ）（定価¥99,800）
P&A超特価.// TEL下さい。

注目!!冬のボーナス一括払い手数料(金利)無料（10月末/11月末/12月末のいずれかをご指定下さい。）

X68000 CompactXVI/XVI/XVI-HD

※送料¥2,000、消費税別

今月の特選// 特価品

■Compact XVI さらにお安くなります。

●CZ-674C-H
●CZ-608D-H
●CZ-6FD5（5"FDD）

定価¥492,600

P&A超特価~~¥320,000~~

（※X68000サービスゲーム全て付いています。）
（モニターをCZ-606Dに変更の場合¥10,000を引いて下さい）

右記セットでお買い上げの方にもれなくプレゼント//
①「ダウンタウン熱血物語(¥8,800)」
はもちろん、さらにその上、人気の
㋑「ロードス島戦記(¥9,800)」
㋺「グラディウスII(¥9,800)」
㋩「ザ・プロサッカー68(¥9,800)」
㊁「信長の野望武将風雲録(¥9,800)」
㋭「ELLE(エル)(¥7,800)」
の中のいずれか2本をプレゼント//

X68000-CompactXVI ●ディスケット10枚●ジョイカード2ケプレゼント中// さらにお安くなります。

Ⓐセット：CZ-674C＋CZ-608D……定価¥392,800▶特価~~¥281,000~~

12回	23,400	24回	12,400	36回	8,600	48回	6,700	60回	5,600

X68000-XVI ▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中//

Ⓐセット：CZ-634C-TN＋CZ-606D-TN……定価¥447,800▶特価価格はTEL下さい。

12回	25,900	24回	13,700	36回	9,500	48回	7,500	60回	6,300

Ⓑセット：CZ-634C-TN＋CZ-614D-TN……定価¥503,000▶特価価格はTEL下さい。

12回	29,400	24回	15,500	36回	10,800	48回	8,400	60回	7,100

X68000-XVI-HD ▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中//

Ⓐセット：CZ-644C-TN＋CZ-606D-TN……定価¥597,800▶特価価格はTEL下さい。

12回	35,700	24回	18,900	36回	13,100	48回	10,300	60回	8,600

Ⓑセット：CZ-644C-TN＋CZ-614D-TN……定価¥653,000▶特価価格はTEL下さい。

12回	39,100	24回	20,700	36回	14,300	48回	11,200	60回	9,400

※上記のモニターを、CZ-606D（定価¥79,800）、CZ-604D（定価¥94,800）、CZ-607D（定価¥99,800）、CZ-605D（定価¥115,000）、CZ-608D（定価¥94,800）、CZ-614D（定価¥135,000）、CU-21HD（定価¥148,000）に変更の場合、TEL下さい。超特価で販売致します。

X68000シリーズ～P&Aスペシャルセット

（送料¥2,000・消費税別）

SUPER-HD
(CZ-623C-TN)
●ハードディスク81MB搭載
●平均アクセスタイム19ms
●SCSIインターフェイス標準装備
●SX-WINDOW Ver.1.0搭載
●メインメモリ 2MB標準

注目 スペシャルプレゼント.//
★SUPER-HDには、上記の①をプレゼント
★PRO-IIには、上記の①+㋑～㋭の中の２本をプレゼント

ズバリ価格で大奉仕中

●ディスケット10枚、●ジョイカード2個プレゼント中

SUPER-HD P&A特選セット ★ハードディスク81MB搭載.//

Ⓐセット：■CZ-623C-TN(単品)……定価¥498,000▶特価¥178,000
Ⓑセット：■CZ-623C-TN＋CZ-606D……定価¥577,800▶特価¥233,000
Ⓒセット：■CZ-623C-TN＋CZ-608D……定価¥592,800▶特価¥246,000
Ⓓセット：■CZ-623C-TN＋CZ-607D……定価¥597,800▶特価¥248,000
Ⓔセット：■CZ-623C-TN＋CZ-614D……定価¥633,000▶特価¥268,000
Ⓕセット：■CZ-623C-TN＋CU-21HD……定価¥646,000▶特価¥278,000

PRO-II P&A特選セット 限定

Ⓐセット：■CZ-653C(単品)……定価¥285,000▶特価¥138,000
Ⓑセット：■CZ-653C＋CZ-606D……定価¥364,800▶特価¥195,000
Ⓒセット：■CZ-653C＋CZ-604D……定価¥379,800▶特価¥197,000
Ⓓセット：■CZ-653C＋CZ-608D……定価¥379,800▶特価¥207,000
Ⓔセット：■CZ-653C＋CZ-607D……定価¥384,800▶特価¥209,000
Ⓕセット：■CZ-653C＋CZ-614D……定価¥420,000▶特価¥229,000
Ⓖセット：■CZ-653C＋CU-21HD……定価¥433,000▶特価¥239,000

X68000用ハードディスク（送料¥1,000 消費税別）

①LHD-FM100E▶超特価（ロジテック）(外付)(¥99,800) TEL下さい。
②LHD-FM200E▶超特価（ロジテック）(外付)(¥138,000) TEL下さい。
③EFX-100B▶超特価（エニックス）(¥118,000) TEL下さい。
④EFX-140B▶超特価（エニックス）(¥138,000) TEL下さい。
⑤HD-J100…▶特価¥ 84,000（システムサコム）(¥128,000)（送料・消費税込み¥ 87,550）
⑥HD-J130…▶特価¥100,000（システムサコム）(¥148,000)（送料・消費税込み¥104,030）
⑦HD-J170…▶特価¥117,000（システムサコム）(¥189,000)（送料・消費税込み¥121,540）

プリンター（送料¥1,000 消費税別）

（ケーブル 用紙 付）

■CZ-8PC5-BK(¥96,800)
▶特価¥68,500
■CZ-8PK10(¥97,800)
▶特価¥71,000
■CZ-8PG2(¥160,000)
▶特価TEL下さい。
■CZ-8PG1(¥130,000)
▶特価TEL下さい。

モデム

■PV-M24B5
（AIWA）（定価¥39,800）
▶特価¥25,000
（送料・消費税込み¥26,780）

■MD-24FB5V
（オムロン）（定価¥39,800）
▶特価¥25,500
（送料・消費税込み¥27,295）

■FMMD-311G
（富士通）（定価¥35,800）
▶特価¥24,800
（送料・消費税込み¥26,574）

P&A特選パソコンラック（消費税別）（送料無料）

①3段¥7,900

1230(H)×600(D)×650(W)
（消費税込み¥8,137）

②4段¥8,800

1230(H)×600(D)×650(W)
（消費税込み¥9,064）

③5段¥12,500

1310(H)×700(D)×640(W)
（消費税込み¥12,875）

全機種＝移動自由(キャスター付)●5段のみ＝キーボード収納可能、電源コード付(2.5m)(2P)

●本広告の掲載の商品の価格については、消費税は含まれておりません。 ●営業時間＝平日AM10:00～PM7:00、日祭AM10:00～PM6:00

アフターサービス万全
全商品保証付。専門の担当がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

★頭金なし!!
★即日発送!!

秋葉原でおなじみのP&Aがズバリ超特価セールでご奉仕!!

●お近くの方は、お立寄下さい。専門係員が説明いたします。
●本体単品でも受付します。詳しくは、お電話にてお問合せ下さい。
●ビジネスソフト定価の15%引きOK!! TEL下さい。
●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

全国通販

X68000用ソフトコーナー（送料1ヶ〜5ヶまで￥500・消費税別）

◆Z's STAFF PRO68 Ver3.0(ツアイト)	定価￥58,000	特価￥37,500
◆Z's TRIPHONY デジタルクラフト(ツアイト)	定価￥39,800	特価￥27,000
◆テラッツォ(ハミングバード)	定価￥19,400	特価￥13,600
◆マジックパレット(ミュージカルプラン)	定価￥19,800	特価￥14,200
◆たーみのる2(SPS)	定価￥17,800	特価￥13,000
◆Mu-1 Super	定価￥39,800	特価￥28,500
◆CMA68K(シティソフト)	定価￥29,800	特価￥21,800
◆サイクロンEXPRESSα68	定価￥98,000	特価￥69,000
◆C-TRACE68 Ver3.0(キャスト)	定価￥98,000	特価￥68,500
◆G68K Ver.2 PRO	定価￥22,000	特価￥17,300
◆C&Professional Pack V3.2(マイクロウェアジャパン)	定価￥80,000	特価￥57,800
◆ウエットペイント1〜3(ウエーブトレイン)〔各〕	定価￥15,000	特価￥11,500
◆マチエール(サンワード)	定価￥39,800	特価￥28,800
◆Windex PRO68(JEL)	定価￥28,000	特価￥20,500
◆CZ-213MSD MUSIC PRO68K	定価￥18,800	特価￥13,200
◆CZ-214MSD SOUND PRO68K	定価￥15,800	特価￥11,300
◆CZ-215MSD Sampling PRO68K	定価￥17,800	特価￥12,500
◆CZ-220BSD DATA PRO68K	定価￥58,000	特価￥40,000
◆CZ-224LSD The 福袋 Ver2.0	定価￥ 9,900	特価￥ 7,400
◆CZ-225BSD Multiword Ver1.1	定価￥32,000	特価￥23,000
◆CZ-243BSD CYBERNOTE PRO68K	定価￥19,800	特価￥15,000
◆CZ-247MSD MUSIC PRO68K〔MIDI〕	定価￥28,800	特価￥20,500
◆CZ-249GSD CANVAS PRO68K	定価￥29,800	特価￥22,000
◆CZ-251BSD Hyper word	定価￥39,800	特価￥29,400
◆CZ-253BSD CARD PRO68K Ver2.0	定価￥29,800	特価￥22,700
◆CZ-257CSD Communication PRO68K Ver2	定価￥19,800	特価￥15,300
◆CZ-258BSD Teleportion PRO68K	定価￥22,800	特価￥16,900
◆CZ-261MSD MUSIC studio PRO68K Ver2.0	定価￥28,800	特価￥21,200
◆CZ-263GWD Easypaint SX-68K	定価￥12,800	特価￥ 9,800
◆CZ-265HSD New PrintShop Ver2.0	定価￥20,000	特価￥15,400
◆CZ-266BSD PressConductor PRO68K	定価￥28,800	特価￥22,000
◆CZ-267BSD CHART PRO68K	定価￥38,000	特価￥29,800
◆CZ-284SSD OS-9/X68000 Ver2.4	定価￥35,800	特価￥25,600
◆CZ-285LSD C-Compiler PRO68K Ver2.1	定価￥44,800	特価￥32,500
◆CZ-286BSD BUSINESS PRO68K Popular	定価￥28,000	特価￥20,500
◆CZ-287SS SX-WINDOW Ver2.0	定価￥12,800	特価￥ 9,800

★ゲームソフト25%OFF!!（一部ソフト除く）

周辺機器コーナー（送料￥500・消費税別）

①CZ-8NS1	定価￥188,000	特価￥133,000
②CZ-6VT1	定価￥ 69,800	特価￥ 49,500
③CZ-6TU	定価￥ 33,100	特価￥ 23,900
④BF-68PRO	定価￥ 19,800	特価￥ 14,400
⑤CZ-8NM3	定価￥ 9,800	特価￥ 7,200
⑥CZ-8NT1	定価￥ 13,800	特価￥ 10,000
⑦CZ-6BE2A	定価￥ 59,800	特価￥ 42,800
⑧CZ-6BE2B	定価￥ 54,800	特価￥ 39,300
⑨CZ-6BE2D	定価￥ 54,800	特価￥ 39,300
⑩CZ-6BF1	定価￥ 49,800	特価￥ 35,800
⑪CZ-6BP1	定価￥ 79,800	特価￥ 57,000
⑫CZ-6BM1	定価￥ 26,800	特価￥ 19,300
⑬CZ-6EB1	定価￥ 88,000	特価￥ 63,000
⑭AN-S100	定価￥ 36,600	特価￥ 26,300
⑮CZ-6SD1	定価￥ 44,800	特価￥ 32,500
⑯CZ-6BN1	定価￥ 29,800	特価￥ 21,500
⑰CZ-6BV1	定価￥ 21,000	特価￥ 15,200
⑱CZ-6BC1	定価￥ 79,800	特価￥ 57,000
⑲CZ-6BG1	定価￥ 59,800	特価￥ 43,000
⑳CZ-6BU1	定価￥ 39,800	特価￥ 28,500
㉑CZ-6PV1	定価￥198,000	特価￥142,000
㉒CZ-6BS1	定価￥ 29,800	特価￥ 21,500
㉓CZ-8NJ2	定価￥ 23,800	特価￥ 17,500
㉔CZ-6BL2	定価￥298,000	特価￥214,000
㉕JX-100S	定価￥ 89,800	特価￥ 44,000
㉖JX-220X	定価￥168,000	特価￥121,000
㉗IO-735XB	定価￥248,000	特価￥152,000
㉘LC-10C1H	定価￥598,000	特価￥459,000
㉙CZ-6CS1(674C用)	定価￥ 12,000	特価￥ 8,900
㉚CZ-6CR1(RGBケーブル)	定価￥ 4,500	特価￥ 3,600
㉛CZ-6CT1(テレビ・コントロール)	定価￥ 5,500	特価￥ 4,400
㉜CZ-6BP2	定価￥ 45,800	特価￥ 33,300

中古・高価現金買取り 下取りOK!!

■まずはお電話下さい。
下取り専用買取り電話▶ ☎03-3651-1884 FAX. 03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送り下さい。

買取り価格…完動品・箱/マニュアル/付属品付の価格です。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。（差額は、P&A超低金利クレジットをご利用下さい。）
●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。
●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、￥1,000,000までお支払い致します。

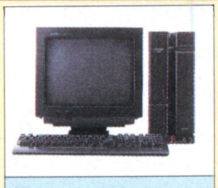
●最新の在庫情報・価格はお電話にてお問い合せください。
●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せ下さい。
●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認下さい。
●本商品の掲載の価格については、消費税は、含まれておりません。
●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

《便利な超低金利クレジットをご利用下さい》

●月々￥1,000円からOK!! ●ボーナス払いOK（夏冬10回までOK）
●支払い回数 1回〜84回 ●お支払いは、8ヶ月先からでもOK!!

●定休日/毎週水曜日

P&A特選＝今月の中古特選品



●CZ-601C
●CZ-611D-TN
¥120,000

●CZ-634C-TN
●CZ-606D-TN
¥248,000

●CZ-644C-TN
●CZ-604D-TN
¥318,000

買取り価格

●CZ-634C	¥170,000	●CZ-602C	¥75,000
●CZ-644C	¥230,000	●CZ-612C	¥85,000
●CZ-604C	¥100,000	●CZ-652C	¥55,000
●CZ-623C	¥138,000	●CZ-662C	¥75,000
●CZ-603C	¥ 85,000	●CZ-611C	¥68,000
●CZ-613C	¥105,000	●CZ-601C	¥45,000
●CZ-653C	¥ 75,000	●CZ-600C	¥45,000
●CZ-663C	¥ 90,000		

下取り交換差額表

新品 / 下取り	CZ-634C モニターセット	CZ-644C モニターセット	モデル UX20セット	モデル CX20セット	9801FA2
CZ-623C モニターセット	150,000	270,000	70,000	160,000	140,000
CZ-613C モニターセット	190,000	290,000	100,000	190,000	170,000
CZ-652C モニターセット	230,000	340,000	150,000	240,000	190,000
CZ-604C モニターセット	180,000	290,000	100,000	190,000	150,000
CZ-600C モニターセット	230,000	340,000	150,000	240,000	200,000

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）
〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
（電信扱いでお振込み下さい。）

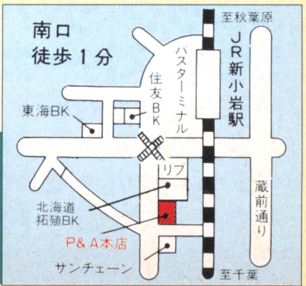
〔振込先〕さくら銀行 新小岩支店
当座預金 2408626 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回〜84回払いまで出来ます。但し、1回のお支払い額は￥1000円以上。

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	3.0	4.0	5.5	5.5	8.5	11.5	16.0	21.0	27.0	33.0

注目!!冬のボーナス一括払い手数料(金利)無料（10月末/11月末/12月末のいずれかをご指定下さい。）



マイコン専門ショップ P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148（代） FAX. 03-3651-0141
営業時間
平日：AM10：00〜PM7：00
日祭：AM10：00〜PM6：00

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

The スーパーファミコン

スーパーファミコンまるかじり!

10/30号

10月16日発売

定価380円(税込)
隔週金曜日発売

全国の書店、コンビニエンスストアにて発売!

別冊付録「真・女神転生」読本

ドラクエV大特集!!

ドワーフの洞窟から、氷の館までを紹介!
「ドラゴンクエストV～天空の花嫁」
ドラクエV裏ワザグランプリスペシャル
[ドキュメンタリー]1992年9月27日ドラクエVを追う!

特集◆最新のソフト情報と購入方法をレクチャーするぞ
'92秋・最新スーパーNES情報

セタが開発した! スーパーカスタムチップの威力

新作FRONT LINE★ファイナルファンタジーV
★半熟英雄★タイニートゥーンアドベンチャーズ
★ミッキーのマジカルアドベンチャー

㊙新作SUPER GUIDE★餓狼伝説★三国志III

BEEP! POWERFUL MEGA-MAGAZINE MEGADRIVE

ビープ! メガドライブ

11月号

好評発売中
定価480円(税込)
毎月8日発売

特集 THEニューフェイス

コナミ、メガドラに参入! セガ・ファルコム、ついにタイトル発表!
そしてソニックの新作は!?

新作ガイド
ソニック・ザ・ヘッジホッグ2
ランドストーカー

MEGA-CD PRESS ▶3×3EYES ▶アネット再び
▶プロ野球スーパーリーグCD ▶ロードブラスター
▶タイムギャル ▶ぎゅわんぶらあ自己中心派2
▶ゆみみみっくす ▶AYA ▶ブラックホールアサルト

ダブル別冊付録
ALL ABOUT TENGEN 総テンゲン色ブック
アレスタの軌跡 コンパイル責任編集 電忍アレスタのできるまで

SOFT BANK
ソフトバンク出版事業部

お近くの書店でお買い求めください

親切と安さの TSUKUMO

冬のボーナス先取り！

'92データショウ発表の新製品もツクモ特価販売中!!

掲載商品２万円以上送料無料(一部地域を除く)

ツクモグローバルカード

大人気！入会者募集中！

18才以上なら学生さんもOK！

国内・外で活躍！使って便利、持ってて安心！ツクモグローバルカードはジャックス・VISAとの提携カードです。ツクモ各店でのお買物がらくらくできる上に、国内はもとより海外での分割ショッピングもOK！
20才以上の方にはキャッシングカードも発行致します。
お申し込みは☎03(3251)9898又は店頭にて！

シャープX68000の事なら何でも揃うツクモにおまかせ！

秋葉原を歩き回る必要はありません。情報が沢山。分らない事は何でもお尋ね下さい。目に優しい10.4型カラー液晶ディスプレイ(LC-10CI)も取り扱い中！詳しくはお問い合わせ下さい。システムのご相談は☎03(3253)1899までどうぞ。

ワープロユースが中心で更にDOS/Vマシンのソフトを使う方へ「書院」パソコンがお勧め

●スーパーアウトラインによる美しい印刷、すぐれた日本語処理能力●ワープロ「書院」の先進機能をそのまま継承●ハードディスク内蔵(Dタイプ)、OADG仕様、DOS/V対応●CPUは32ビット80386SXを搭載。

PC-WD1シリーズ
定価￥333,000より
ツクモ大特価展示販売中！

ツクモTSドライブ X68000用

目のつけどころがツクモでしょ。

TS-XRシリーズ

3.5インチ1ドライブ **TS-3XR1** 定価￥44,800
ツクモ特価￥35,800

3.5インチ2ドライブ **TS-3XR2** 定価￥57,800
ツクモ特価￥46,800

〈仕様〉●3.5インチ2DD/2HD/2HC/1.44MBフォーマット対応
●ユーティリティソフト付属(ディバイスドライバー/フォーマッタ)
※初代X68はROM交換が必要な場合があります。

NEW

5インチ1ドライブ **TS-5XR1** 定価￥53,800
ツクモ特価￥42,800

5インチ2ドライブ **TS-5XR2** 定価￥72,800
ツクモ特価￥57,800

〈仕様〉●Compact XVI用 5インチ2HD/2DD/2HCフォーマット対応
●ドライブ番号切換スイッチ付

※写真はTS-3XR1です。

秋の夜長を大画面で楽しむ

あなたの部屋がミニシアター＆迫力ゲームセンターに変身！

シャープ液晶ビジョンセット
XV-P1 定価￥220,000

今なら「RGB信号→S端子変換ユニット」をサービス！
ツクモ特価￥198,000

ホビーでガンガン使いこなす方へ

10台限り

●X68000の未来を象徴するハイコンパクトなボディ(体積比44%)●成熟するウィンドウ環境、使いやすさと高機能を追求したSX-WINDOW Ver2.0搭載●2HD 3.5インチフロッピーディスクドライブ2基搭載●カラー液晶ディスプレイ接続可能●X68000XVIの高性能を継承●VGAモードサポート(SX-WINDOWのみ対応)

只今、シャープ製ゲームソフト(3本)プレゼント中!!

X68000 Compact XVI セット

●CZ-674C-H(X68000 Compact本体)……￥298,000
●CZ-608D-H(0.28㎜ピッチCRT)……￥94,800
●100MBハードディスク……サービス
合計定価 ￥392,800

ツクモ特価￥348,000(消費税別途￥10,440)
クレジット例(36回払・税込)初回￥14,344×月々￥11,900×35回

電子文具

タイムマネージメントを管理する便利ツール
●従来の電子システム手帳用ICカードがそのまま使えます●次から次へと忙しい方の為の強力な助っ人●大画面・大容量・手書き入力で操作効率向上！

シャープ 電子マネージメント手帳
PV-F1 定価￥128,000

ツクモ特価販売中！

更に、便利な名刺読み取り機「PV-BR1」(標準価格￥120,000)もお勧めです。

パソコン通信

時代は9600ボーへ!!
■モデム 9600bps/MNP5 & CCITT V.42bis ツクモ特価￥54,800～
■通信ソフト たーみのる2 ツクモ特価￥14,000

メモリーボード

■1MB増設RAMボード(CZ-600C専用) ツクモ特価￥19,500
■1MB増設RAMボード(ACE/PRO/PRO2シリーズ用) ツクモ特価￥17,000
■2MB増設RAMボード(拡張スロット用) ツクモ特価￥33,800
■4MB増設RAMボード(拡張スロット用) ツクモ特価￥59,800
※計測技研のメモリーボードも取り扱っております。価格についてはお尋ね下さい。

SX-WINDOWワールド

●SX-WINDOW開発キット **CZ-288LWD** 10月発売
●SX-WINDOW Ver. 2.0 **CZ-287SS** ￥12,800
●Communication SX-68K **CZ-272CWD** 9月発売
●SOUND SX-68K **CZ-275MWD** ￥15,800
●Easypaint SX-68K **CZ-263GWD** ￥12,800

X68000用ハードディスク

お勧めSCSIタイプハードディスク

VIP100CX	特価￥68,000
VIP120CX	特価￥78,000
LHD-FM200E	特価￥108,000
LHD-FM240	特価￥128,000

※SCSIボード(CZ-6BS1 定価￥29,800)は別売です。

X68000用MOディスク

ツクモはSONY MOディスクの正規代理店です。
これが今一番の人気者！
SONY 3.5インチ光磁気ディスクユニットセット
●RMO-S350(3.5″光磁気ディスクドライブ)￥235,000
●SCSIケーブル……￥6,900
●SCSIインターフェースボード……￥29,800
合計定価 ￥271,700

ツクモ特価販売中！

シャープ純正「CZ-6MO1」も特価販売中

スーパーグラフィックセット

その1
WACOM製
●SD-510C タブレット……￥98,000
●TJ-410A-2 接続ケーブル……￥6,000
●SP-200A スタイラスペン……￥10,000
サンワード
●Matier(マチエール)……￥39,800
合計定価￥153,800
ツクモ特価￥128,000

その2
ヒューレットパッカード HP Desk Jet 505J
インクジェットプリンタ……￥99,800
カラーキット……￥12,000
アーベル プリンターケーブル……￥4,800
サンワード Matier(マチエール)……￥39,800
合計定価 ￥156,400
ツクモ特価￥123,000

●JX-220X
A4サイズカラーイメージスキャナ‥定価￥168,000

MIDIセット

NEW Aセット
●SC-55……￥69,000
●SX-68M-II……￥19,800
●Mu-1 SUPER……￥39,800
合計定価 ￥128,600
ツクモ特価￥99,000
(消費税別途￥2,970)
クレジット例(18回払・税込)初回￥6,596＋月々￥6,300×17回

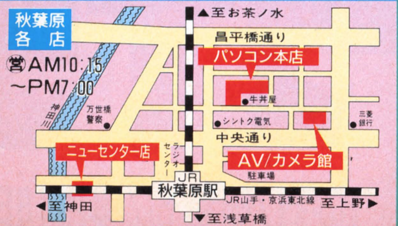
NEW Bセット
●CM-300……￥58,000
●SX-68M-II……￥19,800
●Mu-1 SUPER……￥39,800
合計定価 ￥117,600
ツクモ特価￥92,000
(消費税別途￥2,760)
クレジット例(10回払・税込)初回￥10,919×月々￥10,000×9回)

NEW Cセット
●CM-500……￥115,000
●SX-68M-II……￥19,800
●Mu-1 SUPER……￥39,800
合計定価 ￥174,600
ツクモ特価￥141,000
(消費税別途￥4,230)
クレジット例(15回払・税込)初回￥11,300＋月々￥10,500×14回

冬のボーナス一括払(金利手数料なし!!)受付中。詳しくはお問い合せ下さい。

通信販売のご注文は下記フリーダイヤルへ。
全国どこからでも通話料無料
受・注・専・用フリーダイヤル 0120-377-999
通販センター 03-3251-9911 商品についてのお問い合わせは各店又は通販へ。

クレジット払い	各種リース払い	現金書留払い
月々￥3,000以上の均等払いも頭金なし、夏・冬ボーナス２回払いも受付中！	くわしくは各店にお問い合わせ下さい。ケースに合わせてご相談にのります！	〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係
カード払い	**全国代金引き換え配達**	**銀行振込払い**
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通販部へお申し込み下さい。	お申し込みは☎03-3251-9911へお電話１本！配達日の指定もできます。	事前に☎でお届け先をご連絡下さい。三和銀行 秋葉原支店(普)1009939 ツクモデンキ



ツクモは「スーパーX PRO SHOP」です。

PRO STAFF **ツクモ**

パソコン本店 荒井

九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号
★商品のご注文は在庫確認の上お願いします。★表示価格には消費税は含まれておりません。

ツクモパソコン本店2F
☎03-3253-1899(直通)

パソコン本店代表 ☎03-3253-5599 (担当／荒井) 休毎週木曜

■ツクモニューセンター店 ☎03-3251-0987(担当 沢栄) 休毎週木曜
■名古屋１号店 ☎052-263-1655(担当 山口) 休毎週火曜
■名古屋２号店 ☎052-251-3399(担当 松原) 休毎週水曜
■ツクモ札幌店 ☎011-241-2299(担当 田口) 休毎週木曜
■DEPOツクモ2番街店 ☎011-242-3199(担当 鈴木) 休毎週木曜
※定休日が祝日と重なる場合は営業致します。

PC SPECIAL/ユーザー投票により、パソコン、

本誌が選ぶ,92年

創刊号

話題騒然
読み逃しなく！

●COLUMNS 小田嶋隆、水玉蛍之丞、中尊寺ゆつこらが語るフリーエッセイ●READERS 読者がつくる、秋葉原感覚あふれるページ●PRODUCTS ジャンル別・価格帯別で、あらゆる新製品を分類・網羅●DATA 独自調査のデータベースで現行商品を一望●NEWS 日米パソコン界の最新動向

創刊記念モニター大募集

486パソコン、オーバードライブプロセッサ、ソフトウェアほか

創刊記念プレゼント

PCオリジナルテレホンカードほか

パーソナルコン

月刊

10月17日創刊/毎月18日

周辺機器、ソフトなど92年度のベスト製品を選出

ベストパソコン!

いまユーザーは何を買うのが賢いのか

■Windowsのための、486パソコンのベスト1は?

■ポータブルプリンタ実力ナンバー1の座は?

■使いやすくて表現力豊かなMS-DOSのワープロ王は?

■ヒット商品、ここが良ければパーフェクト!
完璧なDOSマシンだった[PC-9801FA]

■98で組んだMIDIシステムでCDのサウンドを超えた!

■『CD-ROM』で広がるパソコンの世界

■98にオーバードライブを挿す!

■一太郎を使うためのハードディスク環境を徹底整備

■98、Mac、DOS/V、TOWNS、X68000
ゲーム上手で選んだ、ベストパソコンはこれだ!

■USA RUPO いま、米パソコン界では何が起こっているのか
メディアラボに見るバーチャルリアリティの現在

ピュータ総合情報誌

PC

創刊3大付録
1.月刊PC福袋ディスク―Windowsで使う壁紙&アイコン集
2.PC-386NOTE AR活用カード
(エプソン販売)
3.B2判 情報カレンダー
(化成バーベイタム)

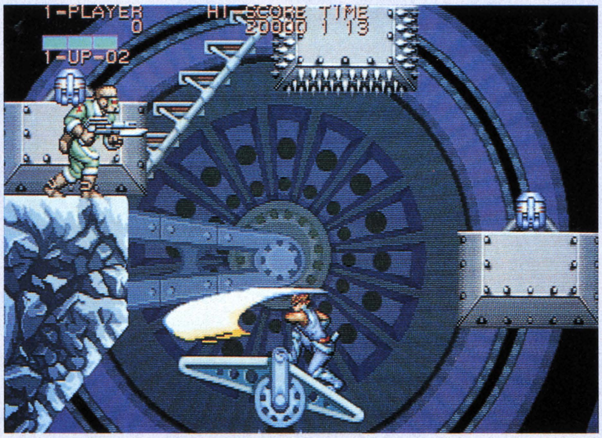
SOFT BANK ソフトバンク株式会社 出版事業部
〒108 東京都港区高輪2-19-13 電話03-5488-1360

発売/創刊特別定価650円(税込)

◆月刊PCでは、パソコンに関するすべてのテーマで投稿を募集しています。宛先は月刊PC編集部投稿係まで
◆創刊号はお近くの書店でお買い求め下さい

SOFTWARE information

前々から噂があったが，カプコンの「スト……」が発売される。そう，「ストライダー飛竜」だ。また，記事には間に合わなかったが，「デスブレイド」がディスク５枚組で10月30日に発売予定となっている。

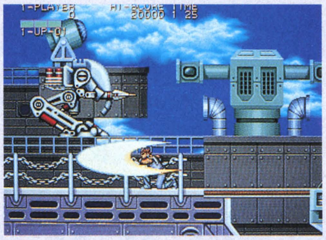


ストライダー飛竜

1989年の春にゲームセンターに登場し，多くのマニアを虜にしたカプコンのジェットコースターアクションゲームの代表作,「ストライダー飛竜」が登場する。

エリート特Ａストライダーである飛竜を操り，人類を支配しようとする冥王グランドマスターを倒すため，彼の手下や悪の機械兵器を精神兵器サイファーで叩き斬りながら突き進まなくてはいけない。赤い国からシベリア，空中戦艦にアマゾン，そして敵の本拠地サードムーンまで，目にも止まらないアクションの連続攻撃には，一瞬のまばたきも許されないだろう。手が，そして指が飛竜とひとつになったとき，無敵の力が宿るはずだ。「貴様らにほかの玩具(ソフト)は必要ない！」。 (八)

X68000用　5″2HD版　価格未定
カプコン　☎03(3340)0718



丘の上のファイナルファイト

1.	ファイナルファイト	1
2.	OVERTAKE	2
3.	ポピュラスⅡ	7↑
4.	グラディウスⅡ	3↓
5.	エトワールプリンセス	ー初
6.	スターウォーズ	4↓
7.	三國志Ⅲ	5↓
8.	出たな!! ツインビー	8
9.	シムアース	6↓
10.	ふしぎの海のナディア	9↓

今月も「ファイナルファイト」が圧勝。なんつーかアレですな，「OVERTAKE」が発売になってくれないと勝負にならないって感じですな。ハガキの中身はというと，スーパーファミコン版よりも忠実な移植度への評価，そして「ストリートファイターⅡ」への要望について書かれているハガキが多いですが，品切れで手に入らないという声も届いてます。

「OVERTAKE」はそろそろ内容が公開されてきて人気もジワジワ上昇してきているようですね。X68000のカーレースものはしばらくなかったので，かなりのヒットが期待できます。画面のセンスもまずまずで，ディテールの凝り方も日本のゲームとしてはかなりのセン。F1ブームもまだまだ続きそうですしねえ。そういやセナは来年どうなるんだろう？

３位にはお待たせ，イマジニアの「ポピュラスⅡ」が上がってきました。前作にハマった人が「あの面白さが帰ってきた」と推薦してくるパターンが目立ちます。グラフィックのパワーアップやあいかわらずの奥の深さ，X68000先行発売などもいい印象を与えています。

初登場５位はエグザクトの「エトワールプリンセス」。「ナイアス」「アクアレス」で好評を博してきただけあって，ユーザーがブランドを信用しているようですね。

ハガキが異口同音に唱えているのが，“女の子がかわいい。こういうのが最近少ない”ということ。この路線に確実に反応するユーザー層ってのがいるんですねえ。

下位のほうが元気がなく，入れ替わりも少なくなっているのが多少気になりますが，また新たなソフトが出てきて順位を賑わせてくれることを期待しましょうか。じゃ。 (浦)

ロードス島戦記Ⅱ

Ｘ68000版「ロードス島戦記」発売とほぼ同時にPC-9801で発表されたのが，この「ロードス戦記Ⅱ〜五色の魔竜〜」である。前作と同じく，大小さまざまなミッションをクリアしつつ，最終的にはメインとなるミッションに挑まなければならない。

パーティに組み入れることができるキャラクターは最大６人で，自分で作ったキャラクター，そしてあらかじめ用意されているNPCのどちらでもOK。そして，この６人のほかに，スペシャルNPCというキャラクターが状況に応じて２人まで加わってくる。パーティは最高８人の大所帯になるわけで，タクティカルコンバットもたいへんな騒ぎになりそうだが，ターボコンバットというのもあるのはありがたい。

Ｘ68000用　5″2HD版　価格未定
ハミングバードソフト　☎06(315)8255

極

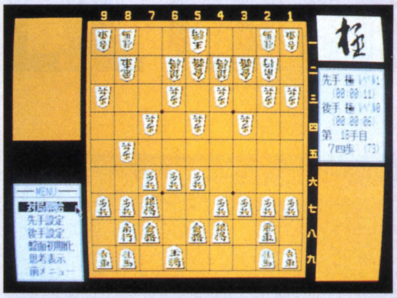
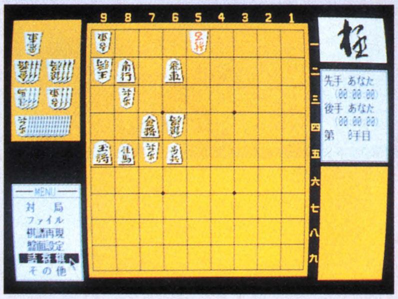
数カ月前に「棋太平68K」と「将棋聖天」という２本の将棋ソフトが発売されたのは記憶に新しい。そして，ここにもうひとつの将棋ソフトがＸ68000で登場する。

この「極（きわめ）」は，“有段者に勝てるコンピュータ将棋ソフトを作ろう”という目標のもとに開発されたそうで，世界最強のコンピュータ将棋ソフトと銘打っている。

思考レベルは，レベル０〜５のノーマルモードとレベル０〜３のハイパーモードがあり，全部で10段階。後者のハイパーモードは有段者コースとして設計されていて，思考時間を度外視して強さだけを追い求めているらしい。

機能もほかの将棋と同じようなものが揃っている。局面の入力，駒落としでの対戦，定跡の組み込み，盤を反転しての対局。もちろん詰め将棋を解かせることもできる。さて，世界最強の言葉は本当なのか？

Ｘ68000用　5″2HD版　12,800円(税別)
ログ　☎03(3837)2595

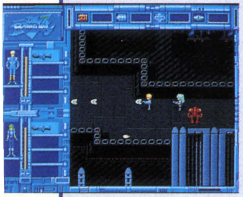
バーンウエルト

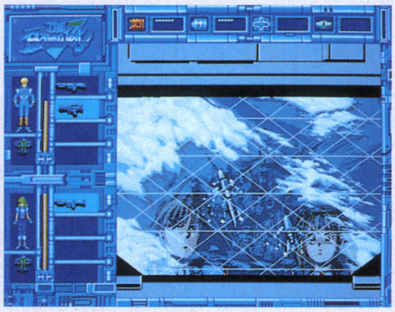
「エメラルドドラゴン」以来，Ｘ68000では音沙汰のなかったグローディアですが，最新作の「バーンウェルト」が発表となります。さあ，穴が開くまで画面写真をトクとご覧あれ。

この「バーンウェルト」は２人パーティで進行する，８方向スクロールシューティングゲームなのです。当然，ひとりで２人動かすのは難しいので，相棒はコンピュータが操ってくれます。２人の組み合わせはストーリーに伴い変化するなどというところはRPGみたいですね。また，グローディアであるから，200枚を超えるビジュアルシーンもあったりするのです。なんだかとんでもなく期待できそうなゲームのような気がするではないですか。わくわく。

順調にいけば10月30日（この本の発売日から２週間くらいあとだね）には店頭に並んでいるはず。来月号ではばっちりレビューする予定なのでこっちのほうも乞うご期待，だ！　（で）

Ｘ68000用　3.5/5″2HD版　9,800円(税別)
グローディア　☎03(3220)5226





エトワールプリンセス

今回はキャラクターの動かし方を中心に。まず，十字ボタンで８方向に動き，Ａボタンでジャンプ，Ｂボタンで攻撃するというのはアクションゲームにはよくあるパターン。で，両方のボタンを押すと，連れてきた味方２人＋主人公リリルの３人の中から使用キャラクターを選択する画面に切り替わる。ここでは所持アイテムやマップを見ることも可能。TOWNSパッドを使っている場合はSELECTボタンで画面切り替え，STARTボタンでポーズとなる。使用するメンバーによって攻撃方法が全然違うので，場面に応じてうまく使わなければいけないようだ。

Ｘ68000用　5″2HD版　9,800円(税別)
エグザクト　☎025(247)9160

TREND ANALYSIS

1992年8月の月間売り上げベスト10

POINT	タイトル	発売元	発売日
342	ポピュラスII	イマジニア	'92/8/28
251	三國志III	光栄	'92/5/28
239	バトルテック	ビクター音楽産業	'92/7/10
216	シュートレンジ	ビッツー	'92/7/24
194	ライジングサン	ビクター音楽産業	'92/8/28
189	キャノンサイト	日コン連企画	'92/5
186	ファイナルファイト	カプコン	'92/7/17
182	シムアース	イマジニア	'92/5/22
148	リーディングカンパニー	光栄	'92/7/30
123	出たな!! ツインビー	コナミ	'91/12/6

今月は「ポピュラスII」が1位になっているが，ポイントを見てわかるとおり，そんなに売れたわけではない。内容的にはなかなかのものなのだが，さまざまな要因が絡んで，いまひとつの売れ行きとなったようだ。

ピーター・モリニューが「ポピュラス」「パワーモンガー」に引き続いて世に送り出したということで話題性はもっているし，移植もがんばっている。対戦に難があるとはいえ，それは実際にプレイしなければわからないことであるから，売れ行きに影響するとは思えない。

はっきりいって，ゲームの内容ではなく，発売日の不明確さ，パッケージの地味さなどが影響したのではないだろうか。

発売日が遅れてしまってわかりにくくなるというのはよくあるが，「ポピュラスII」の場合は“もうすぐ出そうだけど，もう少し時間がかかりそうな気もするなあ”という印象をみんながもっていたところに急に出てしまった，という感じである。

パッケージもきれいな仕上がりではあるものの，アピール度という点においては弱かったと思う。背中の黒地に金色文字というのも，きれいではあるが目立たない配色である。

ただ，前述したように決してつまらないゲームというわけではないので，今後の売れ行きを見守りたい。

さて，初登場は「シュートレンジ」「ライジングサン」「リーディングカンパニー」の3本である。

4位の「シュートレンジ」は最近よくある，お手軽に遊べるシミュレーションゲーム。このテのソフトはファンタジーや未来を舞台にすることが多いようだが，そのほうがやはりとっつきやすいからだろう。

細かい部分の演出は秀逸で，グラフィックもかっこいい。画面上ではなにげないように見えるが，そのつくりのよさはたいしたものだ。

5位の「ライジングサン」は西洋人が作った，日本の戦国時代を舞台としたシミュレーションゲーム。とはいえ，うさんくささはあまり目立たなく，むしろいい雰囲気ともいえる。戦国シミュレーションゲームというと，どうしても光栄の一連のシリーズを思い出してしまうが，まったく正反対の仕上がりになっており，パラメータなどを気にせず，次々と起こるイベント（アクションゲーム）をクリアしていけばいいようになっている。実に海外ゲームらしい内容である。もちろん，どちらがいいかは好みの問題である。

そして，8位の「リーディングカンパニー」は光栄の，“戦国時代ではない”シミュレーションゲーム。近日発売予定にあがっている「エアマネージメント」と同じく，ビジネスシミュレーションである。得意ジャンルとして確定するのか？

さあ，はたして来月は……。

［データ集計協力店］（順不同）
九十九電機本店
J&P（渋谷/町田）
OAシステムプラザ横浜店
P&A
ラオックスGAME館

ウワサのソフトウェア（海外編）

ASHES OF EMPIRE

のっけから「パワーモンガー」を思わせる，重厚だか騒々しいのかわからない鳴りっぱなしの音楽。いい意味でも悪い意味でもAMIGA的だなあと思ってマニュアルを見ると，君の任務は５つの共和国を鎮圧することだ……。というところで，どうもこれは僕の苦手なストラテジーのような気がしてきた。

ゲームが始まると，やはりまずマップが出てきたが，進めていくと予想に反してフライトシミュレータ風の画面が現れた。砂漠のような地形。砂丘また砂丘を越えていくと，戦車もいる。ジョイスティックのボタンを押すとミサイルが飛んでいく。

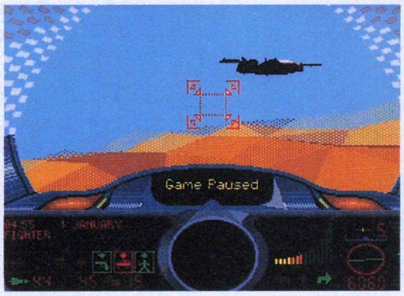
これだけで僕はこのゲームが気に入ってしまったのである。なぜなら，パソコンレベルのフライトものにしてはめずらしく，

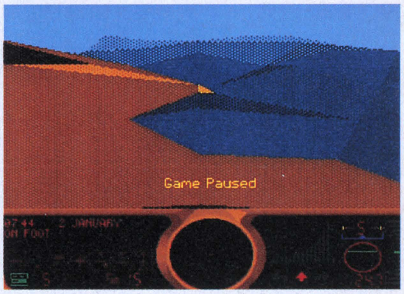
地面に起伏がついている

からである。さらに撃墜されると，**地面の起伏に沿って歩き始め，**そのままさまよっていると海に出たのだが，**海ではポリゴンが波打っていて，海岸に波が打ち寄せていた**のである。

さすがに68000のマシンでは苦しくカクカクするが，68030なら実に滑らかに動く。

AMIGA用のフライトシミュレーションは山ほど存在するが，どれもこれも真っ平らな地面の上に申し訳程度の建物や道や川や山が配置してあるという程度のもの。空高く飛び上がるとディテールは失われ，浮揚感というものが感じられない。唯一飛んでて楽しいのは古典的名作「F-18」。それ以降の作品は結局F-18を超えてない。僕は戦略なんぞに興味はないのだ。





というわけで，ひさしぶりに楽しく飛べそうなソフトに出合えてうれしい。任務そっちのけでしばらく飛び回ることにしよう。ゲームの内容？　戦闘に勝って各所を占領していけばいいようだけど，誌面の都合があるので省略させてもらうわ。ホホホ。

が，最近気になるのは，32ビットでないとつらいソフトも増えてきたこと。スペックからは推し量れない限界の高さもAMIGAの魅力のひとつだと思うのだ。　(A.T.)

発売元　MIRAGE

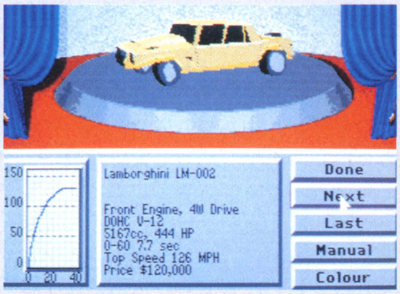
ウワサのソフトウェア（海外編）

4D SPORTS DRIVING

４次元運動運転。なんのこっちゃ。すでに「4D」は日常的な言葉として定着したようだ。空間3次元＋時間1次元，合わせて4次元。動く3次元CGのことを世間では4D CGと呼ぶ。なぜか2次元動画は3D CGと呼ばない。

で，4D SPORTS DRIVINGだ。4D BOXING同様，4D Engine搭載。跳ね橋やループ道路などのさまざまな仕掛けを備えたコースを１周してタイムを競う車のアスレチックゲーム。と，ここまで聞いて不審に思った方もいるであろう。そう，Hard Drivin'にそっくりなのである。

運転していて楽しいゲームというのは，路面をグリップしているという感触がある。「Indy 500」にしろ「World Circuit」にしろ，運転そのものを楽しめるゲームはドライバーの意思が車の挙動に素直に反映する。コーナリングが気持ちいい。直線で馬鹿みたいにスピードを出すのは誰にでもできる。スポーツドライビングの醍醐味はコーナーをきれいにクリアしてパワーをかけつつ立ち上がっていくときの爽快感ではあるまいか。

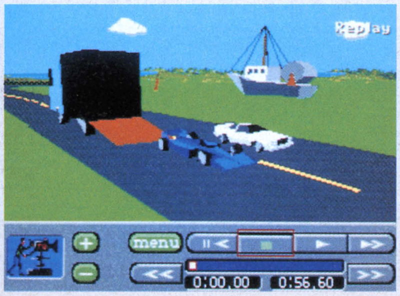


この4D SPORTS DRIVINGは残念ながら運転が面白いとはいえない。特にコーナリングのようにハンドル操作が必要な場面は走っていてつらい。マシンコントロールを失ったというそぶりも見せずに，だらだらとコースアウトする。異様にストレスがたまるのだ。

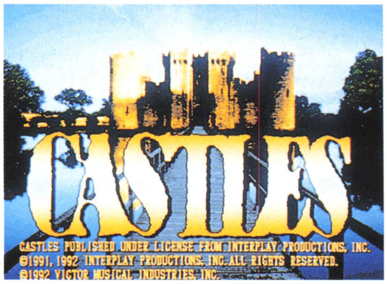
車は多数用意されていて，けっこう作り込んである（跳ね橋を飛び越えるインディカーは奇妙で楽しい）。コースもいろいろあって，コンストラクション機能もついている。ハードディスクにインストールできるなどなど，サービス要素も多く，作りもちゃんとしている。走って楽しいものであればと思うと残念でならない。

試しにコンストラクション機能を使ってIndy 500ばりのオーバルコースを作ってみたが，やっぱり楽しく運転できる代物じゃなかった。ゲームとしては成立していても，なにか大切なものが欠けている気がする。　(A.T.)

発売元　MINDSCAPE



一国八城の主たるべし



またヘンなゲームが出てきた。「シムシティー」も「ポピュラス」も「レミングス」もヘンだが，こいつは輪をかけてヘンだ。インタープレイの作品をビクター音楽産業が移植した「キャッスルズ」である。

Urakawa Hiroyuki
浦川　博之

まずパッケージに書いてあるジャンルの名前がふるっている。「リアルタイム築城シミュレーション」だって。これでゲームの内容がいい当てられる人はいないだろう。わたしゃ，時間内に決められた形の城を作るパズルゲーム的なものを想像していたが，やってみたら全然違うのだった。

世の中に，劉備玄徳の気持ちを味わってみたいとか，F-15とF/A-18を戦わせてみたいという人は多いだろうが，「城を建ててみたい」人はきっと多くないだろうと想像する。だから「キャッスルズ」に世間の注目が集まってるとはちょっと考えにくい。

だけど，マイナーだからこそ声を大にしていいたい。この正体不明の「キャッスルズ」，結構いけますぜ，ダンナ！

築城はデンジャラスな香り

このゲームにもいちおうシミュレートしている年代と史実というものがある。13世紀，イングランドのウェールズ征服だ。時の国王エドワードIII世は，ケルト人の住み家であったウェールズ地方を大胆な戦略によって奪い取ったのである。

その戦略というのがすごい。ケルト人が住んでるその土地に，いきなり出かけていって城を建ててしまうのだ。途中の妨害は武力で跳ね返す。城ができてしまえばこっちのもんで，イングランドから入植者を呼んできてそこに街を作る。城というより城塞都市なわけね。で，あとは住みづらくなったケルト人を追い出すだけってな段取り。大胆というか，なんかもうゲルマン人の文明を振りかざした無茶苦茶な戦略ですな。

この敵陣に城を建てるという作業を行うのがこのゲームの目的であり，プレイヤーの仕事というわけ。与えられたマップの中にできるだけ短時間でできて，攻撃に強い，大きい城を作る。城が完成すればそのマップの征服は完了となる。

しかし，ヤバい。ヤバすぎる設定だ。これを見てケルト人系の方々は怒らないのだろうか。これがもし，「インディアンを押しのけて，アメリカに白人の街を作ろう」という話だったら，絶対に問題だぞ。「満州に日本人の街を作ろう」なんてことになったら，国際問題になることうけあい。あ，あと，これはすごいぞ，「パレスチナにユダヤ人の街を作ろう」。うおー，コンピュータゲームをめぐってテロが起きちまうぜ。ナニ人が作ったゲームかは知らないが，ちょっと「ゲルマン民族の優越感」みたいなものを感じるゲームだったりする。

城の作り方教えます

ゲームが始まると，そこは荒野だった。プレイヤーはマップを見わたして，敵の攻撃から防御のしやすい場所を選ぶ。背後が川だったり海だったりするといいね。水源が近いと堀を作ることもできるし。

場所が決まったら設計コマンドで城をデザインする。使うパーツは塔と，壁と，それから門だ。

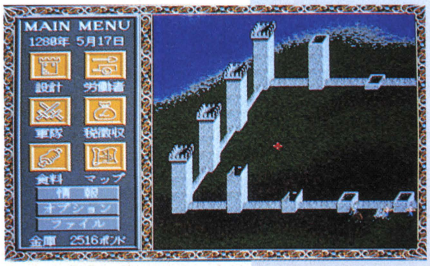
強い城を作るにはちょっとコツがある。壁をずらーっと並べるのは，壮観ではあるが，すぐにだーっと壊れてしまう。これは当時の建築技術を無視するから悪いんで，壁3枚に塔を添えるのが基本なのだ。敵の襲撃の多そうな箇所には高くて厚い壁を作っておこう。お好みによって弓を射るための窓や，壁に張りついた敵兵に油をかける油口なども取りつけることもできる。城の形は適当でOK。資材をなるべく使いきるようにすればよい。

城のデザインができたら，いよいよ建設の開始だ。労働者を雇って，城の各箇所に配置する。おいしい城の作り方。城1カ所につき人夫90人。あとは注いで飲むばかり。いまなら築城デキャンタ差し上げますってなもんだ。

建設が始まると，この仕事ぶりを眺めるのが結構楽しい。馬車引きやら大工やら，職業によってアニメーションが違うのだ。ひたすら1輪車を押してるやつ，足場に立って木鎚をふるうやつ。壁がぴょこっと高くなったりするのを見ているうちに不思議と時間がたつ。

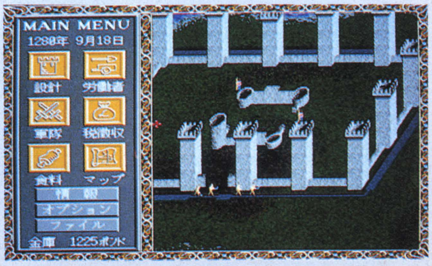
そんなある日。画面の中では壁を作り終

X68000用　5"2HD版2枚組　9,800円(税別)
ビクター音楽産業　☎03(3423)7901



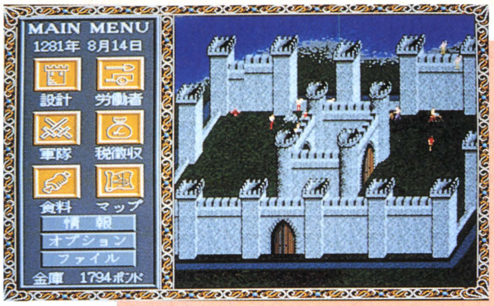
柱がニョキニョキ伸びてくる

のどかだけど命がけの戦争風景だ

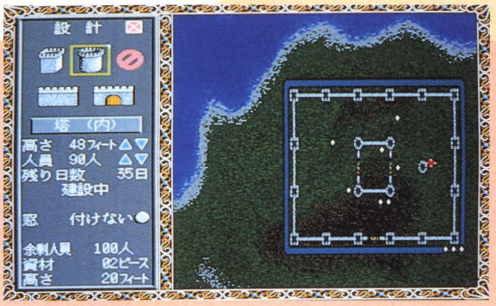


堀を作るのは兵隊さんの役目

えた小人たちが足場を取っぱらっていたのだが，ふと気づくと壁の上に大工がまだ残っている。
「ん……？　取り残されたのかな？」
大工がこっちに向かって手を振り始めた。なんだこいつと思った瞬間，
“ピュルルルルル。ぐちゃ”
大工が壁から身を投げてしまったのである。死体のすぐ隣では，あいかわらず作業が続けられている。
……このゲームを作った人っていったい……。背筋が寒くなった私であった。



この中に自殺者が。さて，誰でしょう



全体の見取図。基本形だ

我が道を阻むもの

「西の方角から敵兵が攻めてきました！」
自分たちの土地にこんなものを建てられてケルト人がだまっているわけはない（ちなみにケルト人以外にも襲ってくる勢力はある)。襲撃には歩兵と弓兵で立ち向かう。凝ったシステムを期待するかもしれないが，これはホントに素朴な戦闘である。塔があれば，そこに弓兵を配置する。歩兵は敵の来る方角に置く。この「敵の来る方角に」というのが結構重要だ。うっかり反対に配置してしまうと，全員城壁にひっかかって，ケルト人が城を壊していくのを見るばかりになってしまう。
だから勝つための最大のコツは，敵の来る方角を見落とさないこと。このゲーム，結構ヒマなのでついついマンガなぞを読んでしまうのだが，気がつくともう敵兵襲来のメッセージが消えていて，泣きながら適当に兵隊を配置するしかなくなる。

帝王の選択肢

「ノーシャー侯爵が湿地帯の戦いに引きずりこまれて往生しております」
ときどき築城の途中にイベントが挿入される。プレイヤーが自分が王様であることを思い出すひとときではある。
「侯爵様は兵士を集めるために100ポンド送ってほしいと申しております。さもなくば戦いは続けられません」
A)　100ポンド送る
B)　ほかの指揮官を任命する
C)　100ポンド送るが，ほかの指揮官を任命する
プレイヤーがやることは，この3択に答えるだけ。築城以外の部分はホントに大ざっぱなのだ。しかし，この3択によって部下同士のパワーバランスや教会との関係などに影響があるのでバカにはできない。お金を持っていかれたり，兵隊を貸さなきゃいけなくなったり，あるいは敵兵が襲ってきたりと，築城にも影響がある。
これら並み居る障害を切り抜けて，王様は必死にお城の完成を目指す。城が大きければ大きいほど兵隊もたくさん雇えるし，税金も集められるので，ちょこちょこっと作ってハイ完成というわけにもいかないのだ。門をつけて完成してから，1回敵の襲撃に耐えられれば合格。次のステージへと進める。ウェールズの完全制覇には，8つの城を建てなければならない。だんだん地形は悪くなるし，敵は強くなるし，城も大きいのを建てなきゃいけなくなるし。

みんな同じではありません

ひと言でいって，結構偏ったゲームだ。本来の領土であるイングランドへのコマンドは一切なし。兵隊の訓練度も，兵法もなし。ステージ間のつながりもないに等しい。もうひたすらに城を建てることにこだわっているのだ。「信長の野望」の，経済社会軍事まんべんなくシミュレートした均整さに比べ，この偏りようはなんだ。
しかも，この偏り方が妙に気持ちいい。いかに速く，効率よく城を作るかに全力を注げるようになっているのだ。申し訳程度に添えた3択のイベントも，このゲームをただの積み木遊びにしないためにいい味を出している。
しかし，有名海外ゲームのくせに操作性があまりよくないというのは，ダンスの苦手な黒人，ホームランの打てない外人選手にも似た情けなさがある。
たとえば城の設計である。設計画面ではマップを縮小して表示している。よって狙ったところにパーツを配置するにはひじょーに細かい操作をしなきゃならない。設計画面なんか作らずに，城なら城のパーツをつかんで通常画面にホイっと置けるようにするのが，マウスオペレーション時代のゲームデザインではなかろうか。
それから，城のデザインを変えてもあんまり機能に変化がないのも少しものたりない。「今度はこういう城を建ててやる」という気持ちを起こさせるものがあれば，もっとハマリ度は上がると思う。個人的には西洋の城の建築に関するサブマニュアルがほしかった。
だが，操作性の悪さなどは，このゲームのコンセプトのよさとインパクトで十分おつりがくる。なんつったって，自分の設計した城がだんだん形になっていくのを見るのがたまらない。画面で労働者が実際に働いているのを見るのが，とてつもない快楽なのだ。自分なりに城に思い入れをもって設計すれば，なんとしても完成させてやるという気合がわいてくる。この日本じゃ絶対に生まれてこない偏ったゲーム，一度はやっておくべきだろう。

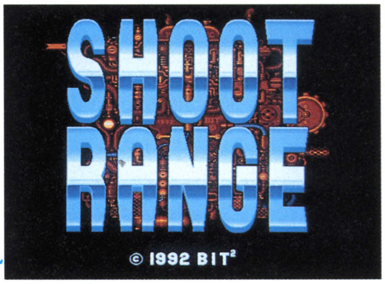
インパクトのあるのは七難隠す

操作性はあんまりよくないし，X68000版のコンピュータの反応もあんまりシャープとはいえない。しかし「キャッスルズ」には「キャッスルズ」でしか体験できない面白さがある。車でいうとイタリア車って感じですか。
いったん労働者が作り出してしまえば，あとは作業が終わるたびに次の持ち場へ回すのだけがプレイヤーの作業なのだが，動き回る労働者と，しだいに完成していくグラフィックを見るのが楽しくて飽きない。突然使者がやってきたり，敵が攻めてくることもあるし。しかしまあ，ノンビリしたときにプレイしないと，たいしてすることもないのに画面を見てなきゃいけないのが苦痛に感じることもあるかもしれない。
海外では続編も発売されているということだが，やがてX68000でも遊べるようになることを期待する。

総合評価（0　5　10）

操作性	★★★★★
グラフィック	★★★★★★★★★★
音楽	★★★★★★
ゲームデザイン	★★★★★★★★★
箱庭度	★★★★★★★★★★

エンジンぶるぶる歯車ガシガシ



Takahashi Tetushi
高橋 哲史

ロボット同士を格闘させろ。6対6の戦いは全滅するまで終わらない。ルールは単純，動きはこまやか。1戦ごとに経験値を重ね，奪った賞金でパワーアップ。スポーツ感覚で楽しめるシミュレーションゲームだ。

最近ふとした拍子にメガドライブを購入してしまいました。前々から“ほしーなー”とは思っていたのですが，財政的になかなかふんぎりがつかず，のびのびになっていたんです。ところがそんなあるとき，知り合いから「5,000円で譲っちゃるぞ」との甘い誘惑が飛び込み，ここぞとばかり「うおーっ，そのメガドラもらったあー！」と奪ってきたわけです。

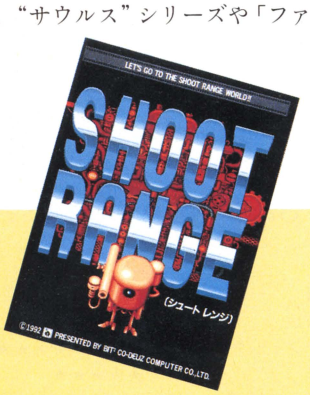
そして気がつけば，手元にはいつのまにかメガドラのソフトが十数本ばかり（食費を削ってソフトを買うやつ……）。

しかし，所有ゲーム機はメガドラにゲームギア。セガの罠にはまっているのかオレは……。思わずTVチューナーパックを通してベアナックル（メガドラの熱いソフトだっ）をゲームギアの画面で遊んでしまうというわけわからんことをやってしまう私だ。でも燃えるーっっ！

BIT²さんいらっしゃ〜い

ということで，全然関係なさそうな前書きから始まってしまいましたが，初参入のビッツーさんです。X68000ユーザーの皆さんにはあまりお馴染みではないかもしれませんが，MSXではかなり「ならした」ソフト屋さんなのですね。

もともと私はMSXユーザーでもありまして，“サウルス”シリーズや「ファミクルパロディック」などで楽しませていただいていただけに，今回のX68000初参入作品「シュートレンジ」にもちょっと期待を寄せてしまうのでした。



X68000用　5″2HD版3枚組　9,800円(税別)
ビッツー　☎03(3479)4558

CALLING PLUGRUNS TEAM!

ゲームの内容はといいますと，ずばり，「近未来ロボットウォーシミュレーションゲームRPG風味」です。ビッツーの誇るセンスのいいグラフィックが目に心地よい画面を作り上げ，期待せざるをえなくしてくれます。それではさっそくゲームの中身を覗いてみましょう。

「シュートレンジ」ではオーソドックスなターン制を採用していますので，相手のロボットが動いては自分のロボットを動かす，の繰り返しでゲームが進行します。全部隊にひと通り命令を出す“大戦略”シリーズと違って，ロボット1体ごとにターンが割り振られていますので，どちらかというと将棋やチェスに近い感覚があります。

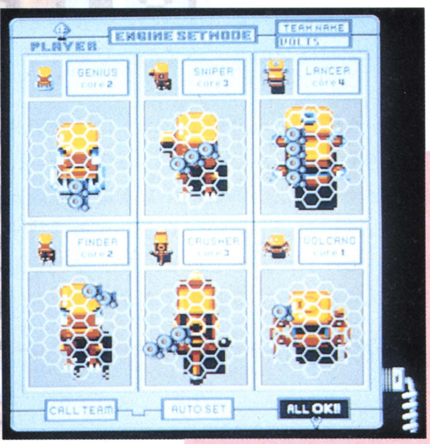
最初に部隊を選びます。同じ性能の部隊がデザインを変えて“NUTS”“VOLTS”“PLUGRUNS”と3種類用意されており，自分の気に入ったデザインで遊ぶことができるようになっています。

まるで棋士が愛用する手駒でしか将棋をしないというような雰囲気を感じるようになっていて，実に好感がもてます。こういう遊びの精神ってのはおしゃれでいいですよね，ほんと。ちなみに私はPLUGRUNSがお気に入りです。

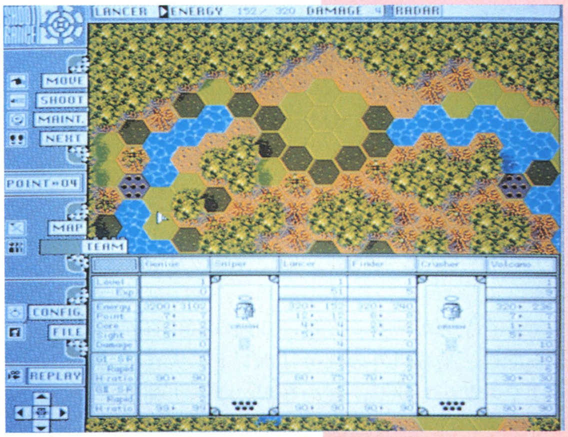
さて部隊を選ぶと，次にそれぞれのロボットにエンジンをセットしていきます。わにゃわにゃと小気味よく動くエンジン（本当に動いてるんだ，これが）をそれぞれのロボットにセットして，いよいよ戦闘開始です。

マップを選んで部隊を配置して，「さあ，敵はいずこ！」と，あたりを見回しても誰もいらっしゃいません。マップをスクロールさせてみても，いろんなメニューを開いても，パッケージを振ってみても，ブリッジして鼻からソーメンを食べてみても，敵はどこにもいないのです。

それもそのはず，シュートレンジでは各ロボットの視界内に入った敵しか表示されないようになっているのです。要するに，“大戦略”シリーズの索敵モードと同じですね。ということでいちばん視界の広いFINDER（戦闘偵察機）を先頭にしながら，じわじわとほかのロボットたちも動かしていきます。



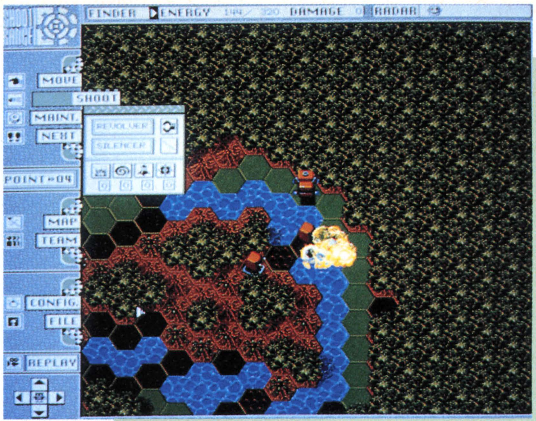
ロボットにエンジンをつけてやる

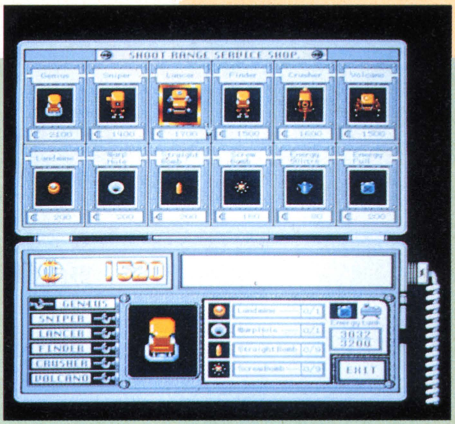


チームの状態を見る。死んだやつには輪っかが……

マップ中央付近で敵のLANCER（高速移動攻撃機）を発見！　さっそくこちらのSNIPER（遠距離攻撃機）で攻撃を仕掛けます。攻撃画面になると敵がアップで表示され，どの部分を攻撃するか尋ねてきます。

この画面もなかなかかっこよくできていてそそるものがあります。エンジンのありそうな場所をマウスで指定して攻撃しますが，見事に的中せず，敵のLANCERは元気に反撃してくれます。

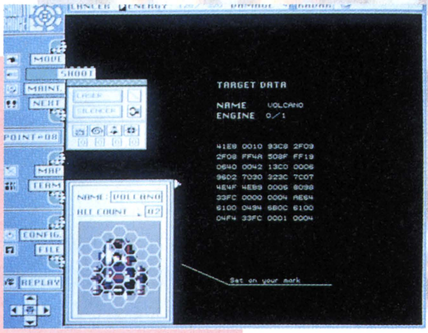

見事撃墜。火の手が上がる



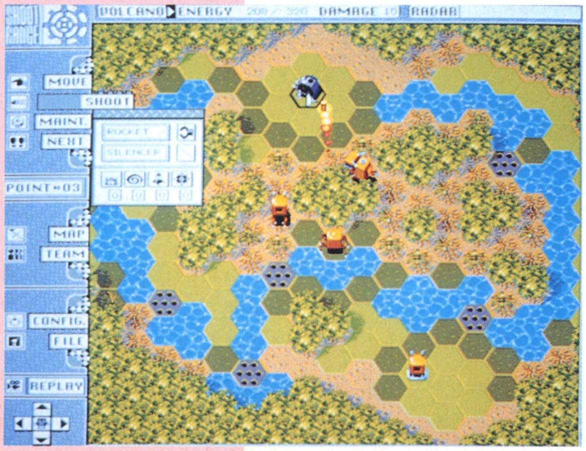
ショップ画面でロボットやアイテムを補充

と，ほどなく自部隊のCRUSHER（接近戦専用機）が煙を上げはじめます。「な，なにぃ相手はいきなりこっちのエンジンを発見したってのかー。サギじゃねぇかー！」などといっているひまはありません。あわててCRUSHERを前線から下げて，GENIUS（修理補給機）に寄せます。ぬぅ，ただじゃおかんぞ，あのLANCER！

さて敵が自部隊の視界で消えたり現れたりを繰り返しながら（向こうから見てもそうなんだろうけど），それでもトムとジェリーよろしく敵を追い回して攻撃を加えていきます。たまに特殊アイテム（地雷，爆烈弾，直進弾）などで攻撃するとすっきりできます。特に爆烈弾（周りにいるロボットにランダムダメージを与える）が見境なくて私は好きです。



エンジンのありそうな場所は？


エンジンを攻撃するためには，まずここで弾を当てねば

最後の１機を撃破したことでめでたく勝利を収め，戦いは終結を迎え……ずに，次のマップに向かう準備のため各ロボットの修理補給が行われます。

また，ここでは与えられた賞金を使ってアイテムの購入などが行えるようになっています。つまりダメージを最小にして勝ち上がっていけば，あとの面での戦いが楽になるというわけです。

頭もフル回転したい

遊んでみて思ったのですが，少しものたりないのです。つまりシミュレーションなのにあんまりシミュレーションしていないのです。どういうことかというと，「あまり頭を使える局面がない」，もしくは「頭の使いがいがない」のですね。

たしかにそれぞれのロボットに特徴をもたせてあり，そのへんの活用を考える楽しみはあるのですが，そこからあまり戦略性が引き出されてこないのです。

要するに，このゲームでは相手チームのロボットすべてを破壊すれば勝利，となるのですが，その肝心要の敵の破壊がほとんど「運まかせ」になっているのです。

前記のとおり，それぞれのロボットには数基のエンジンがセットしてあり，それをすべて壊滅すればロボットを破壊することができます。このアイデアはとてもいいと思いますが，その**エンジンの位置を推理する情報がほとんどない**のです。

いちおうエンジンは直列に配置されることになっているので，最初のひとつを（偶然）見つければ，あとの位置は予想できるようになっているのですが，これではあまり頭を使った気になれません。私の誇る灰色の脳細胞（おいおい）も思わず“おねむ”になってしまいます。

HIT＆BLOWのように攻撃するたびに「エンジンからどのくらいの距離に当たった」とか，そのテの情報が返ってくればもっとやりがいがあるのですが……。

敵の破壊がポイントになるゲームだっただけに，この点が非常に残念です。まあ，その分，気楽に軽快に遊べていいことはいいんですけどね。

しかし，初めてX68000で作ったゲームとは思えないほど，グラフィック，プログラムが充実しているのも事実です。このあたりはさすがというべきでしょう。特におしゃれな画面，動きのセンスなどには唸らされました。

あと対戦モード（RS-232Cと画面交代の２種類があります）もあるのですが，これは熱くなれるかもしれません。

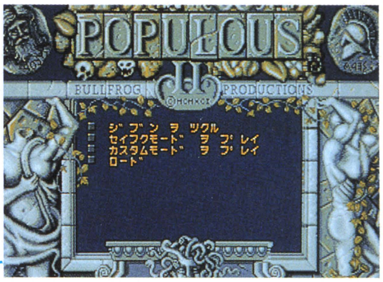
あ，いい忘れていましたが，ゲームをするためにはメインメモリ２Mバイト要ですので，ご注意を。

センスは十分！

本文でも触れたように，グラフィックはたいへんよくできています。キャラのちょっとしたアクションから，戦闘時の描写，メニューをオープンするときに回る歯車にいたるまで，ちょっとしたところでも丁寧に動かしてあって好感がもてます。動きには相当のパターン数を使っていると感じました。今度はこんなキャラたちが画面狭しと動き回るアクションゲームが見てみたいですね。

総合評価	0　5　10
グラフィック	★★★★★★★★★
操作性	★★★★★★★
ゲーム性	★★★★★
ＢＧＭ	★★★★★★
とにかく動く	★★★★★★★★★

神々との戦い 第2章



Nakano Shuichi
中野 修一

買ってくるなり12時間以上遊び続けて周りを呆れさせた中野氏。最近は対戦リーグの準備に向けて燃えていたようですが……。とりあえず，今回は１人プレイの基本テクをまとめてもらいましょう。

予告では対戦ポピュラスIIの企画を行うはずだったのだが，数々の実験と検討の結果，

不可能

と判断された。

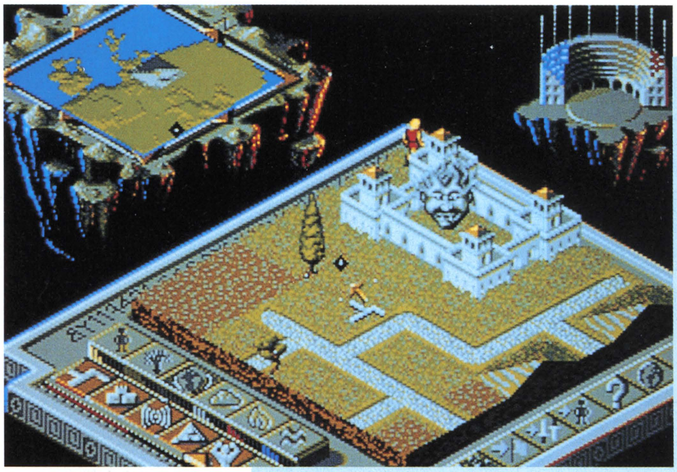
ポピュラスが対戦でよく落ちたのは皆さんもご存じだろう。ポピュラス大会のときも落ちまくりだったが，セーブデータを元に復帰できたのでなんとか対戦を続けることはできた。しかし，ポピュラスIIでは対戦まわりの機能が大幅に縮小され，加えて落ちやすいわ，データの取りこぼしはあるわで，対戦に使用することはほぼ不可能だ。いわゆる「いんこんぱちぶるらんどすけえぷ」のチェックが甘く，２台のマシンで別々のゲームが進行してしまうのである。

＊　＊　＊

ということだが，やはりポピュラスとなると黙っていられない。対戦テクニックがダメなら，征服モードでの基本テクを解説するとしよう。

私も，まだやり込んでいないのでポピュラスIIの総合的な難易度は未知数である。倍の1000面があるとはいえ，最初はかなり手加減しているようだ。ポピュラスで初めて負けたのが49面，ポピュラスIIで初めて負けたのが195面だったから，ポピュラス経験者としてはやや物足りないかもしれない。どうやら100面ごとに関門のあるタイプと見た。

現在250面付近で中断したままだが，まだシンボル移動を使用したアクティブなスプログは行っていない。ポピュラスの第一の極意は神速のスプログである。前作で効率よく領土を広げるために開発したのは，敵領土に対しやや斜めの海上にシンボル移動して直線的に領土を伸ばし，両側に一気に広がる方法だった。



道路はあまり使わない

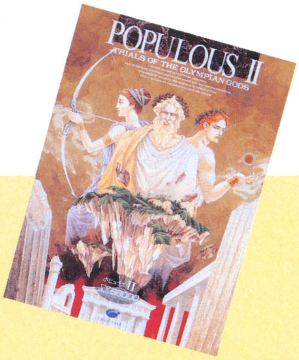
周辺から自然発生的にじわじわ広げていく方法より効率がよく，とにかく海を敵に渡さずにすむという利点がある。欠点は，初期移動でとてつもない速度を要求されるので，やたら難しいということだ。秒単位で最初の２つの城を作り，「テントから出てきた人間がこの地形だと何歩歩いたら死ぬか」を感覚的につかんでおく必要があった。

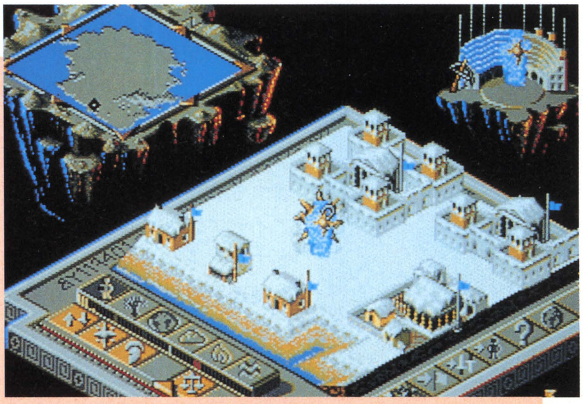
ポピュラスIIでは右クリックによるスプログが可能なので初期造成が爆発的に速くなった。この操作の巧拙で戦況が左右されるので，あまり余計なことを考えるひまがない，ともいえる。人口増加の律速段階が取り払われた現在では，シンボル移動によるスプログはデメリットのほうが大きいのかもしれない。まぁ，いいや。どうしようもなくなったら試してみよう。

ヒーローの使い方

次に気づいたのは，攻め込まれてもかなり敵の攻撃をしのげるということだ。「ヘレン返し」に代表されるように，無敵のヒーローも「？」マークで監視し，雷で狙い撃ちしたり，地割れに落としたりでたやすく迎撃できる。敵はリーダーに集中しがちなので，地震が使える場合はリーダーを攻め込ませて，領地内で殺す。地割れはあまりいじらないこと。自分の領地で地震を起こすのはためらわれるかもしれないが，これが最高の楽勝パターンである。敵のシンボルを壁で囲むという手もある。コンピュータ相手の温泉技（海に沈める）も不可能では，ない（連射マウスがほしい）。

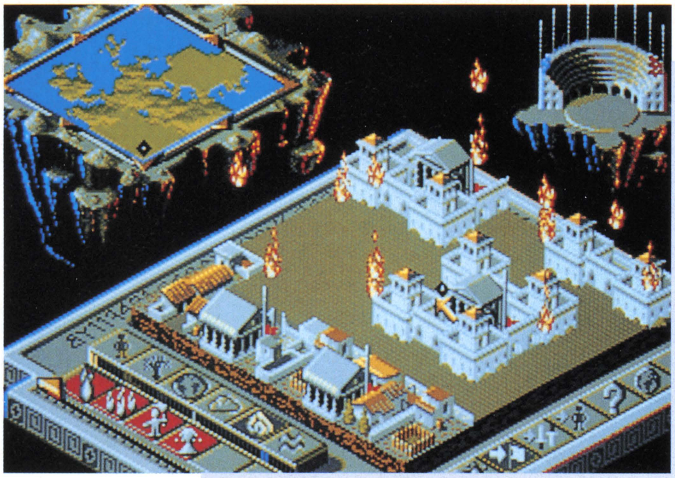
このように基本は「ヒーローの確実な迎撃」となる。逆にいえば，アドニス（どんどん分裂して広がるので狙い撃



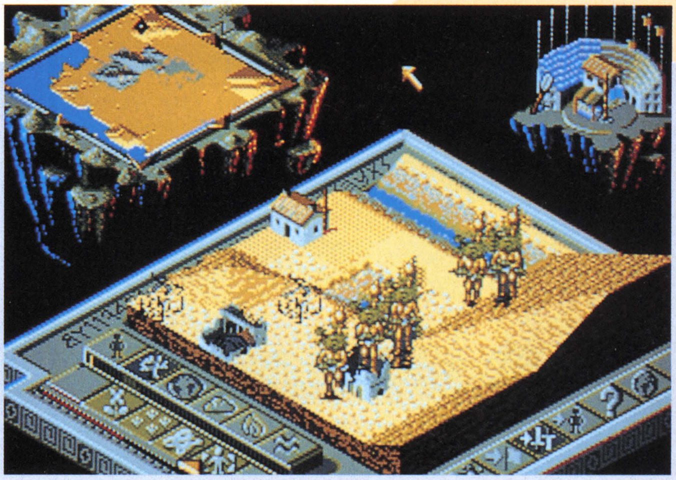


右クリックのスプログはゲームを変えるか？

X68000用　5"2HD版5枚組　12,800円（税別）
イマジニア　☎03(3343)8911



火の雨で城を焼きつくそう



どんどん増殖するアドニス

ちできない）を使うことの有利さも示していることになる。

すべてのヒーローが作れる場合，どれを使用するかという問題がある。簡単に特徴をまとめてみよう。

●ペルセウス（人）

もっとも利口者である。

●アドニス（木）

先ほどいったように，狙い撃ちされない。攻撃効果も大きい。ただし，馬鹿である。地割れなどをまるで避けないので，事前に敵地に悪さをしてはいけない。

●アキレス（火）

関係ないが，間違えてアポロンと呼ぶことが多い（ほれ，サイボーグ009の）。火をつけて回る。事前にたくさん木を生やしておこう。でも，あまり効果はないように思える。

●オデュッセウス（風）

足が速いのはアキレスだろ……という気もするが，とにかく足が速い。敵にやられて初めてその効果を思い知ることになるだろう。神話では利口者のはずだが……。

●ヘラクレス（土）

体力２倍で非常にお得。すなわち量産可能。足は遅いので狙われがちか。

●ヘレン（水）

直接戦わないので体力は低くても量産できる。ヘレンに魅せられた者は，たとえヘレンが滅んでも呆け続ける。とにかく量産すべし。かなりお買い得。水星の守護を受けているので（？）水の上でも歩ける。つまり温泉技はきかないということ。無論，頭もいい。

＊　＊　＊

結局，アドニス，ヘレン，ヘラクレスといったところが扱いやすいようだ。といっても，限られた条件内ではヒーローの果たす役割は同じであり，それほど大勢に影響はない。

その他の神業

●火柱

基本中の基本。火は水で消す。人間相手に使うときは３，４発まとめて使うと効果的（ああ対戦モード）。

●カビ

カビの動作はライフゲームのアルゴリズムである。Oh!X読者なら，小さなパターンで最大の効果をもたらすのはＲペントミノだろうと気づくと思う。これは途中のパターンが破壊されないという条件つきだが，かなり有効な手であることは間違いない。実戦を考えると，長期間生き残るＲペントミノなどよりも，瞬間的に拡大して消えていくパターンやグライダー派生パターンのほうが利用価値は高いかもしれない。対戦ならグライダーで撹乱なども楽しそうなのだが……。いずれにせよ対コンピュータでは奇襲はまったく効果がないのが残念だ。

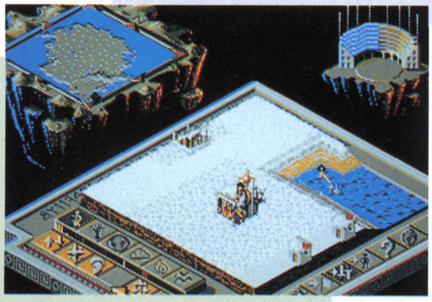
ちなみに，グライダー銃は事実上作成不可能か，ほかの手段に頼ったほうが手っ取り早いと思われる。

●病原菌

かなりやっかいな代物。マニュアルには感染者は殺せとあるが，もったいないので，集合モードで感染した建物をすべて壊し，(可能ならば)ヒーローとして一掃するのが望ましい。

●火山

強力。あまりにも強力。絶対やらせないように。ポピュラスIIの場合，マナが極端に減らないので，10発くらい食らっても反撃は不可能ではない。とはいうものの，非常



雪の中の一軒屋でなにかが起こる

ポピュラスIIに夢中ざんす

とりあえず，「ポピュラスII」は面白い。ゲームスピードには文句ないし，操作性もバツグン，ゲームバランスも絶妙。前作をやり込んだ人にも，前作をプレイしていない人にも，均等にオススメできる。

しかし，これだけ完璧に面白いゲームだと，なんか細かいことをいろいろといいたくなってしまうもの。

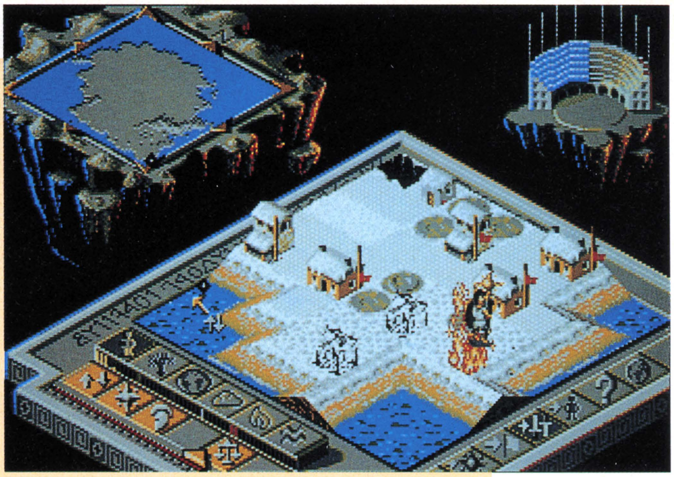
まず，カナメッセージはやめてもらいたかった。雰囲気ぶちこわしだ。「イイモノ」VS「ワルモノ」では，まるで幼稚園児のゴレンジャーごっこではないか。

それと，画面が24kHzのディスプレイモードに対応していない点も気にかかる（いまどきエッチソフトだって対応しているのに）。ゲーム画面は少しでも大きいほうがいい，というのがゲーマーの永遠の願い。できることなら，なんとかしてほしかった。

あとは音楽がなんか妙に浮いてるな。前作では低音が聞いて，「どみゅいーーん」という重々しい印象が大きかっただけに，今回みたいな旋律重視の軽めの曲はなんだか違和感がある。

効果音もＸ68000のAD PCMが単音であることから，効果音がかち合って何が鳴っているんだかわからないときがあるし。ま，これはしょうがないか。

いろいろ，いってみたけれど秀作であることにかわりはないし，面白さは私が保証するからぜひ買って遊んでみてね。秋・冬のゲームはこれで決まり。　（善）



丹氏のポピュラスII神業エディタ

に苦しい。発生したら同時に盛り下げて火口を封じること。大道芸として，火口は雷槌，花などで埋めることもできる。なお，火口の真ん中に木を生やすということもできる(火は止まらない)。山はただちに平坦にして，火口の火柱があるうちに地面を焼く。焼け地は上げ下げで緑地に戻る。

●津波

火山は沈まない。

●沼・地震・泉

敵地内のヒーローがまっすぐ自分の作った沼や地割れに向かっているときは雷で沼などを焼くこと。

どのパラメータを増やすべきか?

ポピュラスIIは経験値制なので経験値(最大5つ)をどのように配分するかが問題となる（各，16×16＝256段階)。

考えてみよう。局地攻撃技である地震や沼，泉などはそれほど大規模でなくてもある程度効果を上げることができる。経験値が上がっても効果の変わらないものも多く存在する。効果の上がるものとしては，雷，嵐，火柱，火の雨，渦巻きなどだ（カビの動きも速くなるらしい)。

特に，マナをあまり消費せずに敵を確実に倒せる雷は重要だ。最初は1本か2本の雷槌がビービーいっているだけだが，フルパワー時には雷雲から無数の雷槌が地面に降り注ぐのだ。快感。

私なりの結論としては，

風＞火＞土＝水＞木＞人

となる。森と火の合わせ技や非局地攻撃の地震などは好みによるだろう（フルパワーの地震も凄いぞ)。

なお，対戦では経験値が考慮されるのかどうかなどは未確認である。征服モードの進み具合で使える神業が変わるとあるが，反映させる方法が（マニュアル通りにしても）よくわからない。そこで丹氏が「ポピュラスII神業エディタ」を作ったのだが(写真参照)，結局対戦自体に無理があり使わずじまいになってしまった。

ポピュラスIIの謎

ポピュラスII第1の謎は得点である。アーマゲドン連発技は使えなくなっているし，普通にやって勝ちすぎると経験値がもらえない。どうやら，神業をたくさん使い，神業をたくさん食らうことが高得点への道らしい。これまでの最高点は65136というものだが(トロフィー5本)，どうしてこんなに取れたのかよくわからない。地震と火柱などでいじめ抜いた結果である。次いで62946点(トロフィー4本)。このときは，敵の大技（津波）を食らいながら，ヒーロー全種類量産攻撃＋アーマゲドンで決着という派手な一戦だった。

特殊コマンドという怪しげなものもある。現在確認されているのはAMIGA版と同じ「MUSIC ON/OFF」だけだが，怪しい。

しかし，最大の謎はやはり対戦モードである。対戦モード，デバッグしてないほうに100円賭けてもいい。あれ？　L誌で対戦やってる！　マップを指定して対戦？　いったいどうやったんだろう……。

対戦の手順

どうしてもやってみたいという人のために正しい手順を示しておこう。

まず，双方に公平な対戦用のマップを作成することが必要である。

2台のマシンを用意し，リセット後，カスタムモードで対戦接続する。

対戦にはそれぞれでマスターディスクが必要となっているが，スニーカーネットワークが構築できる場合なら1枚でもかまわない（要するに手渡しする)。同時に1CPUでの使用という条件は記載されてなかったと思うので大丈夫だろう(かな？)。さらにいえば，必要なのは特定のファイルだけなので……（以下略)。

対戦が始まったら地球アイコンでモード選択に戻り，ペイントマップを選択。ペイントマップでは，点対称な位置に人が配置されている地形が出てくるまで根気よくマップを更新する。運よくそのようなマップとなったら，最初の家ができる前に素早くF10キーを押しポーズさせる。

これからマップのエディットである。まず，F8で陸地を消滅させ，それぞれのリーダーの部分のみ2段以上盛り上げる。ポピュラスIIのペイントマップは海面から1段の高さのものは対称にしてくれないので注意。

以下の操作は，片方だけで行う。F7キーを押し，対称化する。このときリーダー両者が離れているようならポーズを解き，シンボル移動と集合モードで同じ位置に誘導する。同じ位置になったらまたポーズすること。

あとは好きなように地形をエディットする。

マナの量を調整する。いったん最低に戻し1段階上げるとよい。

画面をリーダー中心の位置に戻す。ポーズを素早く解除し，地球マークでメニューに戻る。ゲーム条件を揃えたら，ペイントマップを解除してセーブする。

今度はそのファイルをロードし，ゲーム開始直後にメニューに戻り，「イイモノ/ワルモノ」を双方入れ替えて，もう一度ペイントマップに移行する。ペイントマップ起動の瞬間F10キーでポーズする。ワルモノのリーダーをマップに表示したら，ポーズ解除し素早くメニューに戻る。ペイントマップを解除しセーブする。

「非常に素早く」というのがコツで，絶対に人を平地近くに置かないことが望ましい（家を作らないように)。ペイントマップ以外でポーズすると動かなくなるので注意すること。

これでマップは完成。前作のマップ作りの100倍くらい難しい。このマップを使用するときは，対戦モードで一度メニューに抜け，双方でロードすること。

銀河の必殺救助人見参

Yaegaki Nachi
八重垣 那智

銀河のありとあらゆる宝物を手中に納めようとしている宇宙海賊「キャプテン・KERO」。その次なる照準が地球に向けられた。近未来戦士「フォックス」を操って，次々に現れるボスキャラを倒し，人類の平和を守のだ。

叩け宇宙海賊

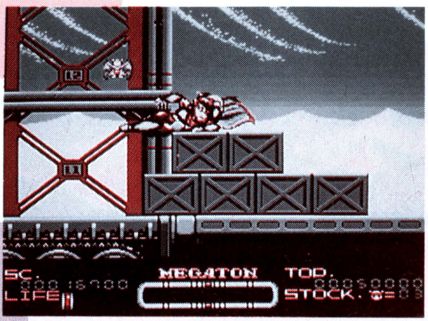
この「サンダーレスキュー」は新鋭ソフトハウスのギミックハウスが開発した，戦闘型ジャンプアクションゲームである。発売はブラザー工業のTAKERUから。デビュー作として期待が高まるが，まずはゲームの概要を紹介していこう。

はるか未来において，銀河を征服せんとし，地球の侵略を行う宇宙海賊KEROの野望を，主人公フォックスを操り，打ち砕くというものである。全体としてはナムコの「ベラボーマン」のような印象を受けた。

レバーとボタンの組み合わせで移動や攻撃を行い，攻撃はため射ちをすることで，メガトンパンチという貫通ショットを発射できる。ジャンプは押している時間で滞空時間を調整でき，しゃがんでいるときにジャンプを押せばスライディングになる。キック攻撃と移動を同時に行え，しかも歩くよりも速く移動できたりするのが特徴である。

ゲームの展開は敵の待ちかまえるフィールドを突破し，最後までたどりつけばボスと対決になる。ボスを倒せばステージクリアとなる典型的なタイプである。

ちょっと変わっているのは，最初の５面はセレクトによって好きな順番にプレイができることであろう。それをすべてクリアすると，さらに２面あり，結局は全７面の構成になっている。



スライディングのほうが歩くより速い

地球を救うために

ゲームの舞台は街の中や森といったところで展開されていく。最後にはKEROの待つ要塞でと舞台のバリエーションはなかなか豊富である。それぞれの面のボスは，いずれも特徴的な攻撃を仕掛けてくるので，動きを読んで，攻略パターンを作らないと歯が立たない感じである。そういった意味では難易度もそれなりで，攻略のしがいはあるのかもしれない。

しかし，このゲームはこういった表現では出てこない問題点が山積みされている，といわざるをえない。具体的に例を挙げてみることにしよう。

まずは操作性である，主人公の動作がままならないのである。状況によってジャンプの反応が異なり，なおかつ勝手に自分の向きが変わってしまうようなことも起こる。これではせっかくの攻略パターンを作っても，それを駆使することは難しい。実際にやっていると，ここのところは非常に気になるのである。

またエリアの途中の敵が貧弱であることも，ゲームを冗長にしている印象を受けた。場面にかかわらず，攻撃もしてこないような敵しか出てこないのでは，いてもいなくても同じに思えてしまうのである。唯一の例外は最終面だが，ほかの面は基本的に色が違ってカタいだけで，何も変わっていないのと同じとしかいいようがない。

戦いの虚しさ

エリア中のカプセルもごく一部を除いてただの得点アイテムだから，はっきりいってこのゲームは，ボスとの戦いしかゲームとして成立していないに等しい状況である。しかし，それすらも操作性の不備により，希薄な内容になっているのは残念という以前の問題なのかもしれない。もっともっと違う部分にも努力して，おもしろい作品を作ってほしいのだが，結局こんなことしかいえない自分がちょっと悔しい気もするのである。

１面のボスはカブトムシ

X68000用 3.5/5″2HD版2枚組 4,800円(税込)
ブラザー工業(TAKERU) ☎052(824)2493

ムネン，アトヲタノム

個人的には，ジャンプアクションゲームはほかのアクションゲームなどに比べると，苦手に感じてしまいがちだ。別にゲームの都合で，人間が軽々と自分の身長の何倍もジャンプするのが納得いかないなどというのではなく，比較的こういったゲームでは，正確な操作をこと細かに要求されるからかもしれない。都合のいい考え方をすれば，小さいことにこだわらない大きな心を持っているということなのだろう。ものはいいようである。

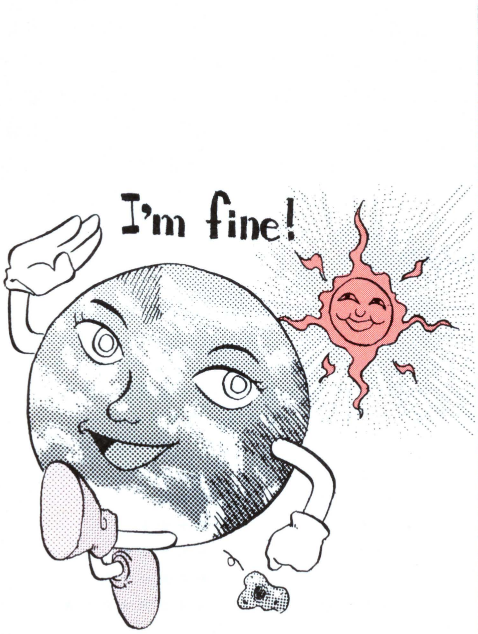
アイデアとしては，目を引くものもいくつかあったので，これからの作品に期待したいところ。こういったアクション中心のゲームがいっぱい出てくれることは，個人的にもうれしいことなのでぜひ歓迎したいところだが，あんまり出すぎてレビューを100ページも書かされたら，嫌なので程々がいいなあ，うんうん。

総合評価 (0 / 5 / 10)

ゲーム性	★★★
技術	★★★★
サウンド	★★★★★
グラフィック	★★★★★★
最終面難易度	★★★★★★★★

AFTER REVIEW

待望のSX-WINDOW対応ゲームソフト「シムアース」。これのために，SX-WINDOWやメモリを買いに走った人もいるようですね。今後のアプリケーションの発売への期待も込めた声がたくさん寄せられています。



シムアース

▶ガイアが気に入った。
深澤　達(14)東京都

▶ゲームとしてみるより，純粋のシミュレータとして感動できる。ゲームとは何か，また，シミュレーションとは何か，を体現している。
中村　守男(23)京都府

▶進化してくれると，とってもうれしい！自分も地球と「一緒に生きた」って感じになれる。
池永　智哉(20)福岡県

▶僕は昆虫が文化を持った姿を思い浮かべて，鳥肌が立ちました。
木原　直也(18)茨城県

▶私は恐竜にナノテク文明まで到達させました。エクソダスはもうすぐだ！
宮岡　三幸(25)神奈川県

▶画面の狭さが身にしみる。XVIに買い替えたくなる。
滝川　直樹(30)大阪府

▶ガイアを爆発させて太陽系第３惑星を消し去りたい。
山田　俊英(24)東京都

▶７月号の記事にうながされたわけではないのですが，とうとう増設RAM(XVI用)を購入しました。遅いけど，RAMディスクにコピーしてやれば……うるさくなるだけだった。ところで，J・ラブロックはシムアースをやったことがあるのでしょうか。
高橋　哲也(20)岩手県

▶おもしろい。SX-WINDOW上で動くAIIIも欲しい。
渡部　裕亨(25)佐賀県

▶２Mバイトだと，これ以外何もできない。早く，「４Mバイトに進化しました。ポペピポピー」となりたいものだ。ところで，気圧を下げる方法はあるのか？　誰か教えてくれ！
鈴木　勅彦(23)愛知県

▶シムアースを買ったのだが，SX-WINDOWがなくて，２週間できなかった。
宇野　寛和(16)東京都

▶たくさん売れれば，きっとバージョンアップしてくれるだろう。
笹山　和宏(24)愛知県

▶私の場合は「死ぬ明(あ)〜日(す)」ですが……ね。
堂領　輝昌(18)神奈川県

▶「地球」という大きいスケールのものを，みごとにまとめてある。
増田　秀樹(26)東京都

▶地球の表情がおちゃめ。
板垣　央(18)千葉県

▶竹内均の監修で，『Newton』にも広告が載った。
竹鶴　敏夫(18)広島県

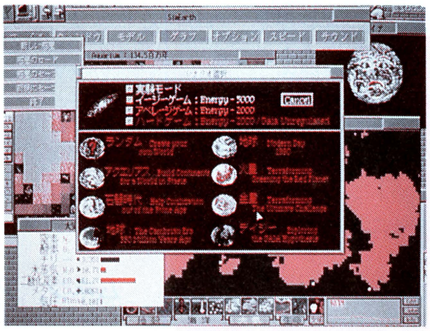
▶すごい。
小宮　崇(21)埼玉県

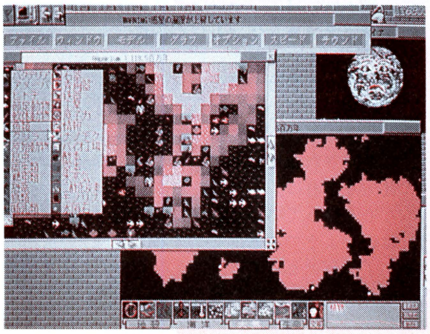
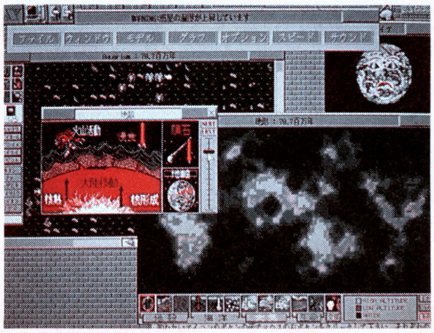
▶スクロールオンの状態で，もとの表示範囲外の部分の画面表示が乱れてしまうのだが……。
深沢　享広(20)東京都

▶いろいろ問題はあるが，やはりSX-WINDOW上で使用できる(私は10Mバイト)ことが魅力です。
須田　浩章(31)埼玉県

▶暇がなくてもゆっくりと遊べる。
葉山　鏞一(21)北海道

▶パッチ当ても行われ，なんとか使えるようになった。
松尾　文人(23)京都府



▶メモリとコプロを買ってきたのはいいけど，ディスプレイが病院送りでまだバックアップもとってない(とれるんでしょ？)。
平田　賀一(17)岡山県

▶パソコンでしかできそうにない感じがいい。　吉田　周理(19)埼玉県

▶ゲームのようでゲームでない。こんなソフトがもっとあればいいのに。
高橋　竜一(21)茨城県

▶えらい！　錦織　信幸(20)山梨県

▶アクション派には絶対受け入れられないだろうな。年をとるにつれてこういうゲームが好きになっていく。これがパソコンのゲームだ！　亀田　峰之(19)香川県

▶まだまだ道は遠いが，早くたのむよ。
遠藤　正彦(21)岐阜県

▶環境ソフトだ。　川井　健司(31)埼玉県

▶シミュレーションゲームはあまり好きではないが，このゲームにははまった。もう少しスピードが速いともっと楽しいと思う。
笠井　博一(22)静岡県

▶コンセプトがよい。ウィンドウシステム対応(しかし，少し値段が高い)。
宮武　克昌(25)神奈川県

▶ずいぶんと待たされたが，なんといってもSX-WINDOW！　期待に違わず，マニュアルプロテクトでキーディスクプロテクトなし，ハードディスクへのインストールも自由自在。心配された速度も，背景同時書き換えなしにすると，ちょっと我慢すればいいというくらいにはなるし，実際モデルウィンドウの作業に手がかかることを思えば，支障はない。でも16MHzだともっと「快適」なのだろうな。
一倉　昭好(24)神奈川県

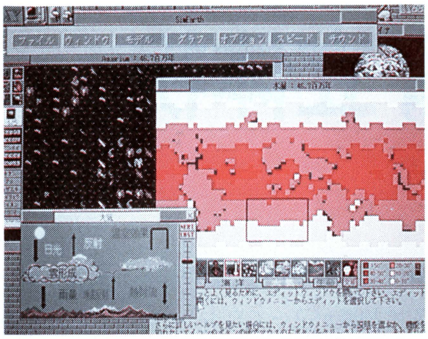
▶24kHzモード(いわゆる−G18オプションを付けて起動したやつ)ですると，ラスタースクロール(？)を始める。Oh!DANGER!
宮内　大輔(17)大阪府

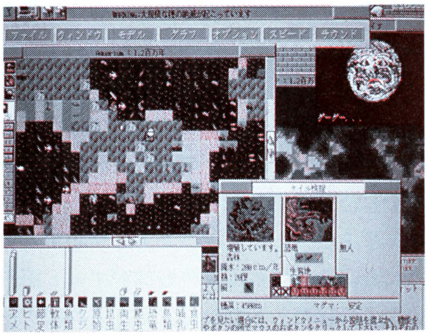
▶SX-WINDOW上で動くのがよい。これでもっとSX-WINDOW上で動くアプリケーションやゲームが出てくればいいのだけれど。　河端　邦春(24)愛媛県

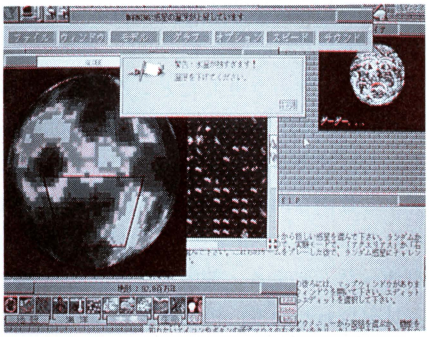
▶お店に予約したけど，メモリ不足で使えないことがわかってしまった……。
平　泰治(17)東京都

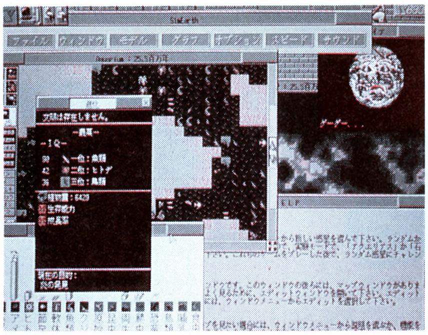
▶すぐ止まってしまうところが，いけず。
鬼沢　徳一(22)茨城県

▶X68000版を見て，ガイア君があんなに恐ろしい顔になっていたので驚きました。
黒武者　健一(22)神奈川県









▶一度神様になってみたい。
大峠　茂(22)千葉県

▶自分独自のものが作れる。
鈴木　幸太郎(21)静岡県

▶マニュアルを読むだけで楽しい(特に作者のつぶやきが……)。
宮岡　三幸(25)神奈川県

▶みんなで買い支えてSX-WINDOWアプリケーションを増やすんだぞ！
森下　寛和(20)鳥取県

発売中のソフト

★**ネクタリス**　システムソフト
X68000用　5″2HD版　7,800円(税別)

★**ふしぎの海のナディア**　ガイナックス
X68000用　3.5/5″2HD版　14,800円(税別)

新作情報

★**キャッスルズ**　ビクター音楽産業
X68000用　5″2HD版　9,800円(税別)

★**バーンウェルト**　グローディア
X68000用　3.5/5″2HD版　9,800円(税別)

★**ヴェルスナーグ戦乱**　ファミリーソフト
X68000用　3.5/5″2HD版　9,800円(税別)

★**チェイスH.Q.**　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版2枚組　7,800円(税込)

★**エトワールプリンセス**　エグザクト
X68000用　5″2HD版　9,800円(税別)

★**沈黙の艦隊**　ジー・エー・エム
X68000用　3.5/5″2HD版　12,800円(税別)

★**オーバーテイク**　ズーム
X68000用　5″2HD版　9,800円(税別)

★**エアマネージメント**　光栄
X68000用　3.5/5″2HD版　11,800円(税別)

★**スクエアリゾートハイパー戦車戦**　ファミリーソフト
X68000用　5″2HD版　4,500円(税別)

★**バトル**　ジー・エー・エム
X68000用　3.5/5″2HD版　12,800円(税別)

★**デスブレイド**　SPS
X68000用　5″2HD版　価格未定

★**餓狼伝説**　ホームデータ
X68000用　5″2HD版　価格未定

★**究極タイガー**　KANEKO
X68000用　5″2HD版　価格未定

★**ファルディア**　M.N.M Software
X68000用　5″2HD版　価格未定

★**ドラゴンスレイヤー英雄伝説**　SPS
X68000用　5″2HD版　価格未定

★**鮫！　鮫！　鮫！**　KANEKO
X68000用　5″2HD版　価格未定

★**TATSUJIN**　KANEKO
X68000用　5″2HD版　価格未定

★**エアバスター**　KANEKO
X68000用　5″2HD版　価格未定

★**サバッシュII**　ポプコムソフト/グローディア
X68000用　5″2HD版　価格未定

★**倉庫番リベンジ/ユーザー逆襲編**　シンキングラビット
X68000用　5″2HD版　価格未定

★**ストライダー飛竜**　カプコン
X68000用　5″2HD版　価格未定

★**ロードス島戦記II**　ハミングバードソフト
X68000用　5″2HD版　価格未定

★**極**　ログ
X68000用　5″2HD版　12,800円

いよいよ本題，立体視

Ogikubo Kei 荻窪 圭

女の子のモデルは見つからなかったけれど，とりあえず立体視はやります。立体に見えないという人もいるでしょうが，じっと見つめいれば……。でも，あんまりやりすぎて目を回さないようにしましょう。

ええい，何いってやがんだ，ってもう遠い昔の話になっちまったが，先月書き忘れたのだからしようがないぞ，高校野球。

ややこしくて胡散臭い建前がどうだとしても，だ。自分たちがスター選手の活躍を願っているからって，それを当事者たちに押しつけちゃいけない。ファンタジーと現実の区別がつかないなんて，大人として恥ずかしくないのかね。

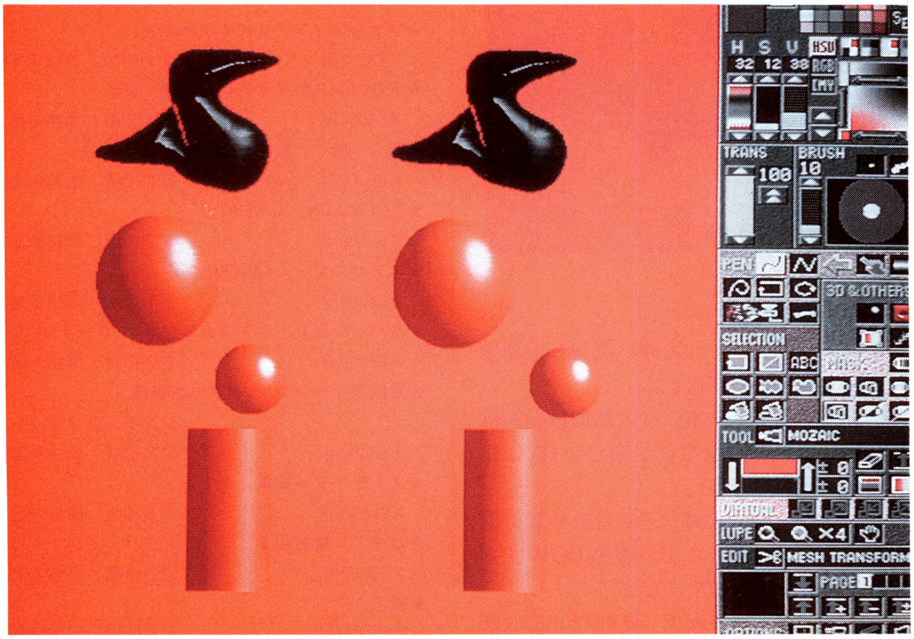
だいたい，何の話かっていうと，あの明徳うんちゃらって高校の連続敬遠的四球攻撃のことだが，なんで意図的四球だと非難を浴びて，バントならチームプレーって褒められるんだ？

そうでしょう。打ちたいのに打たせてもらえない。勝負したいのにさせてもらえない。勝つために誰かが犠牲になる。それをチームプレーだと賞賛してきたのは，誰なんだいったい。

それが，スター選手が絡むと，すぐこれだ。ホンットにアホ。敬遠も立派な自己犠牲，バントも立派な自己犠牲，スクイズなんて神様が泣いて喜ぶわい。教育的だなあ，ホント。そんなに，爽やかとかいう幻想を味わいたいのなら，ルールを変えちまえばいいじゃないか。ちょっとくらいローカルルールがあったってかまわないだろう。金属バットのときみたいに。敬遠を禁止，あるいは，意図的な四球を禁止する，なんていってさ。ついでに，バントも禁止するようにね。そうすれば高校野球がずっと面白くなるのはたしかだから。そうすればチームプレーだってもっと意味のあるものになるだろうに。

なんて，すっきりしたところで話は変わって，9月22日付の朝日新聞にX68000が出ていた。メガネ屋さん用のシステムは実はずっと昔からあったのだが，朝日新聞に載ったのは，インテリアデザイン用のX68000を使ったシステム。部屋のグラフィックに家具を合わせたり，模様を変えてみたりして，部屋の様子をシミュレーションしてみるものだそうだ。というわけで，報告でした。

図1　表題「訓練用プリミティブな立体視」



裸眼立体視の話の続き

そういえば，昔ゲーセンに立体視F1ゲームがあったねえ。筐体の上から立体視用の液晶シャッター付きゴーグルを下ろして目の前において立体感のあるレースを遊ぶ，っていう。

なんていう名前だったか忘れたけど，どうして忘れたかといえば，まったくもって面白くなかったというだけであって，なかなか残念である。あんまり立体には見えなかったし。動いているものって立体に見えづらいし。

で，先月の図がうまく立体に見えたかというと，なかなか見えなかった人もいるのではないかと。ってのは，いろいろと書き込んだりした補助線とかがじゃまをするからだ。だから，図1は先月のやつをおまけ。基本的なやつ。訓練用といっていいかもしれない。

立体でないものを立体に見せる。

立体視のコツは，なんというか，身体で覚えてください，って。

普通，視線ってのは，笑点に向かって笑う，じゃなくって，焦点に向かって集中する。目と目と焦点で三角形を形づくるわけだ。しかし，平行法では，焦点で視線が交差しては困る。両目で同じものを見てしまうから。左と右とで別の映像を捉えるには，焦点を遠くへずらしてやらなければならないわけで，CRTに向かって裸眼立体視するときはまず遠くに視点を向けるよう努力して，そののち，少しずつ合わせていく，って感じになる。っていうのは私の方法で，人それぞれらしい。

まあ，図2のような感じだということで，

そうなるようなイメージを作っていくわけやね。

よく，4つの絵が見えて，真ん中の2つを重ね合わせて3つにすると，その真ん中の絵が立体に見える，って書いてあるけど，それは，図3のようになっている。目には左目に2つ，右目に2つの絵がちゃんと映っているわけで，その中の左目で見た左側の絵(図ではL1)と，右目で見た右側の絵(図ではR2) を一致させれば，いいのである。たぶん。

脳味噌は偉い，というか，バカっていうか。

脳味噌は偉いついでに，究極の裸眼立体視，っていわれているかどうか知らないが，ランダムドットという類の絵がある。一見，ランダムに点が並んでいるだけなのだが，立体視すると，模様が浮き出てくるという仕掛けになっているのだ。

ランダムドットを立体視するのはけっこう難しい。ピタっと左右の絵を合わせるには，どうしてもある程度意志の働きが必要だからだ。

実際にやってみるとわかるが，左と右がうまく重なる瞬間は，コントラストが強いところなど，どこか絵の目立つ部分に注目している。その注目している部分を基準に合わせてしまえば，全体像が見える。というわけで，我々の常識を無視した絵ほど立体視はやりにくい。立体に見えたときのイメージが湧きにくいのだ。目ではなく脳で見るのだから，いろいろな影響を受けやすいのである。

ランダムドットはただ点が並んでいるだけだから，非常に難しい。

しかも，優秀なランダムドットになると，絵が1枚だけなのだ。左右に分かれていない，1枚の黒い点の集まり。

実はうちの近所に，非常に質の高いものを食べさせてくれる中華料理屋があって，しばしば行ったりするのであるが，その店の壁に貼ってあったのがそのランダムドット野郎ポスターであった。

ただ，ランダムに見えるパターンがA0くらいのポスターの一面を埋めており，上のほうに3Dうんちゃらと書いてある。ずっとなにかヘンだなと思いながらも，どう対処していいかわからず放置しておいたのだが，ランダムドットらしいとあたりをつけたらもう黙ってはいられない。先日，ちょうどそのポスターの前に座る機会があり，挑戦した。

なにか基準がないと裸眼立体視はしづらい，という話であって，このランダムドットポスターの場合には，下の端に黒い点が2つ並んでいる。これを基準にして視線を裸眼立体視モードへ固定してから，ポスターを見てください，っていう親切だ（最低限の親切，って感じだけど）。

で，凝視すること数分，見えてきたのは，立体に波打ったり，渦をまいたりするパターン。「をををを，きたきたきたぁぁ」って，つい興奮してお茶をこぼしてしまったが，まあそれはいいとして，ちょっと気を抜くと立体視モードからノーマルモードへ視線が戻ってしまい，渦や波がランダムな点の集まりに堕落するところなんてのが諸行無常の響きがあってなかなかよい。

ではMATIERで

簡単なところから，図4。これは見てわかるとおり，目である。目。私の目をビデオから取り込んで，増殖させてみた。見つめても催眠術にかかるわけでもサブリミナ

図2　裸眼立体視のやり方

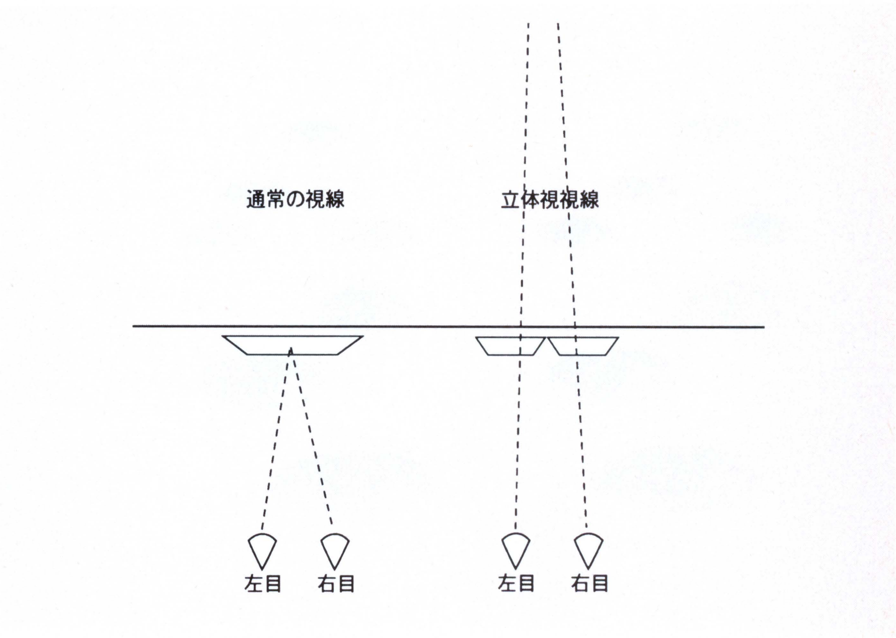
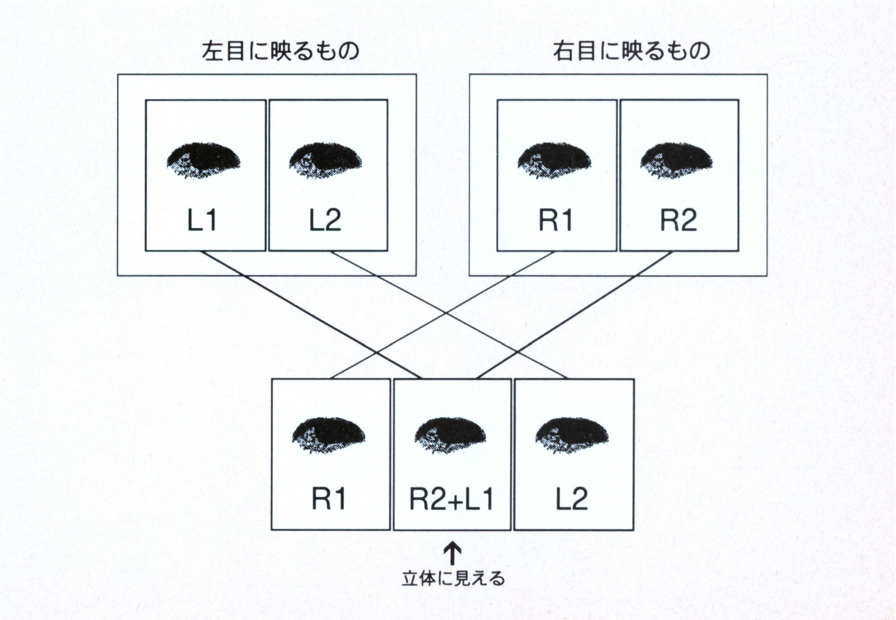


図3　視線を交差させて4枚の絵を見る



ル効果を狙ったわけでもないので，安心して立体視してみてほしい。ちゃんと，下の大きな目が近くに，上の小さな目が遠くに見えれば正解。

まあ，先月のサンプルの実写版ってなところで，不気味なとこ以外はなんの変哲もない。顔のアップから目だけを切り取り，いっぱい拡大/縮小コピーをして，まず右半分を作る。それでもって，左側にそれらをコピーする。

あとは，手前に持ってきたいものは中央寄りに，後ろへ遠ざけたいものは端寄りに動かしてやるだけだ。まあ，お手軽。わかりやすいように，いちおうヘンな配置はしていない。

続いて，メッシュ変形に挑戦。これが図5。とある公園で撮った映像なわけで，これを立体的にするっていったら，やはり遠近感を強調するくらいしかすることがない。ところが，もともと遠近感が強調された映像だったりして（広角で地面すれすれで撮ったから）あまり面白くない。だから，ちょっといじってみたわけだけど，気づかなければ気づかないでいいや。失敗例ね。

失敗のままでもしゃくだから，図4と図5を組み合わせたのが図6。アホだね。アホ。「見つめる公園」とでもしておこうか。立体視すると目玉が迫ってくる感じが出て，面白いといえば面白い。

えっと，まず図4の背景部分をマスクでペイントし，細かい調節をしてマスク反転をする。

それでもって，裏画面に退避しておいた公園の絵を重ねる。重ねるにあたって位置調節のためにちょっと目の位置をずらしたりした以外，作業は単純だ。ごちゃごちゃと細かいことをしようと思うと「MATIER」のアイコンってわかりにくいけど，オンラインヘルプがしっかりしているから，非常に楽であった。

いやあ，オンラインヘルプって，いいもんだわさ。ちなみに，“？”アイコンをクリックするとヘルプモードに入り，左クリックしたアイコンのヘルプが表示される，という寸法になっている。

あとは，色と道具の組み合わせ方を覚えれば，超強力なツールである。

図4　表題「目いっぱい」

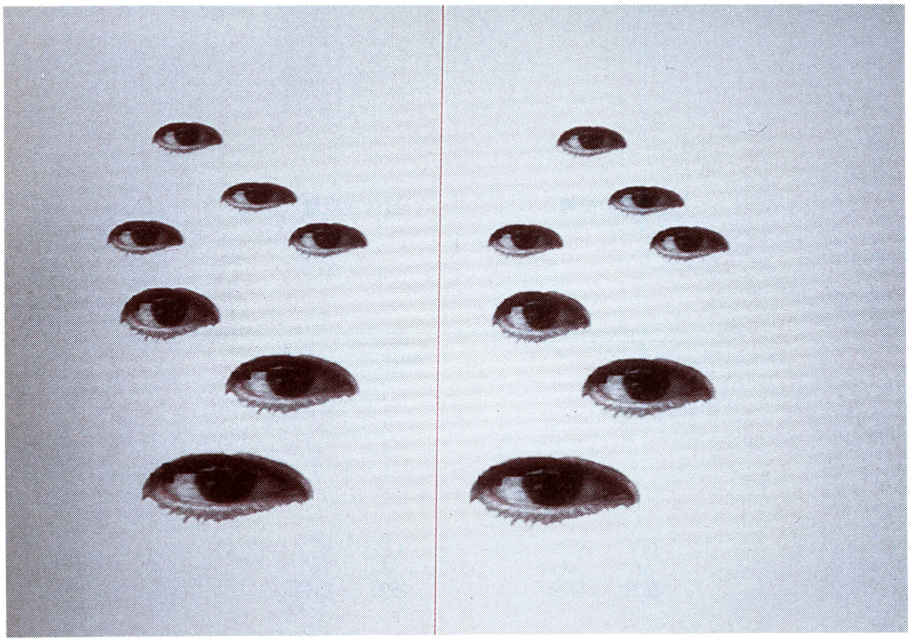
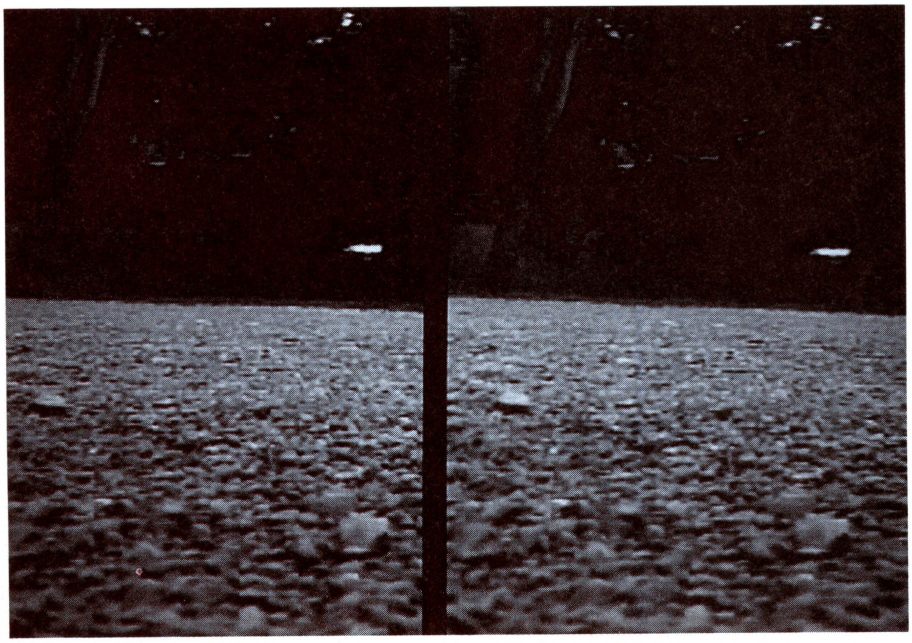


図5　表題「午後の公園」



MATIERの裏画面

「MATIER」はたくさんの画面をメモリに任せて抱えてしまうってのが非常においしいところで，その構成は図7のようになっている。

表画面に対して，1～4の裏画面とひとつのアンドゥ画面があるわけで，特殊効果は表画面と裏画面の合わせ技で，なかなかこの手のツールではサポートしてくれないお遊びを楽しむことができる，ってのがポイントだ。

そもそも，「Z'sSTAFF」とは出自が異なるツールで，そこが最大の魅力であり，「Z'sSTAFF」が古くなってしまった点をうまく補っている。

いうなれば，「Z'sSTAFF」は画面に点を打つためのソフトからスタートしたため，どれだけバージョンアップを重ねても，結局，ルーペでドット修正ってのを強いるような性格を持っている。ver.3.0でたしかにずいぶんと進歩をしたけれども，あくまでも「ドット打ちサポートツール」という前身は隠せない。ドット打ち，線描画といったパソコン用グラフィックツールお家芸が中心にあって，周りを特殊効果ものとかアンチエリアシングものとかがサポートする，

っていう構造だ。

「MATIER」はそうではなく，お絵描きするツールである。パソコンらしいドット打ち的な機能もさまざまな特殊効果的な機能もまったく同じレベルで配置されている。「Z'sSTAFF」に慣れていると最初はとまどうかもしれないが，いざ使ってみると，かなり考えられたメニュー構造になっていることがわかる。独特のルールをもっているが，そのルールさえわかってしまえば，何をするにもいちいちウィンドウを開かねばならない「Z'sSTAFF」に比べて，非常に簡単で俊敏な操作が可能だ。

私はひそかに，これのMacintosh版がほしいな，と思っていたりするくらいで。

メッシュ変形で遊ぶ

本当は若い女の子の裸かなにかあればいちばん遊べたのだが，そういうわけにもいかず（だから，誰かモデルになって，ってば），しゃあないので，昔このコーナーで登場してもらったバルキリー君のディスクをひっぱりだすという情けなさ。

図8のようにメッシュをいろいろとかけて，変形してやる。でもって，前へ出したいところは前へ，後ろへ引っ込めたいときは後ろへ，ってやると，見事，立体的になったりする。

でもまあ，我々の頭の中に「遠近法のルール」がすでに確立されているからより立体に見えやすいというのもあったりして，あまりデタラメに変形をかけても，立体に見えるという保証はない。

今回は立体視ごっこに終始した感があったりするのだが，円筒マッピング技とか，「誰でも描ける秋の草原」技とか，加算合成技とか，いろいろあるし，立体視ごっこでもありがちな，ロゴを浮かび上がらせるとか，前後を逆にして，近くにあるはずのものを遠くに見せてしまうお遊びとかいろいろあったのだけれど，とりあえず今回は，ここまで。

では，みなさんもいいものができたら，「荻窪圭のバカ立体視コーナー係」までよろしく。

来月はなんか思いついたら「MATIER」で遊ぶし，思いつかなかったら……どうしよう。おろおろ。

図6　表題「見つめる公園」

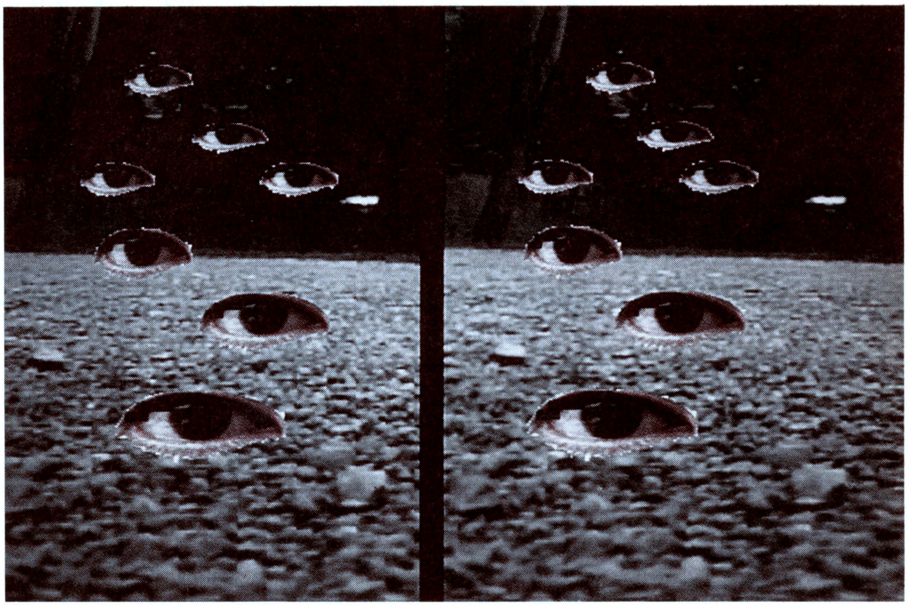


図7　MATIERの複数画面の構成

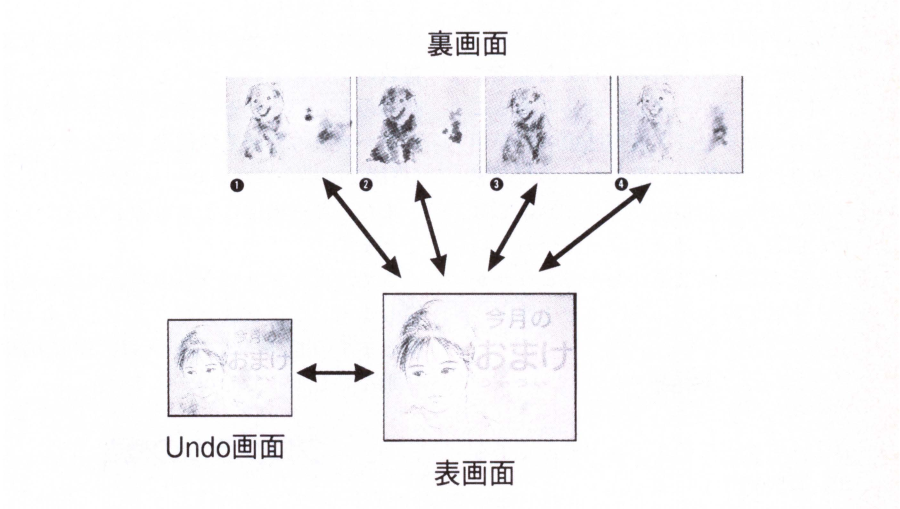
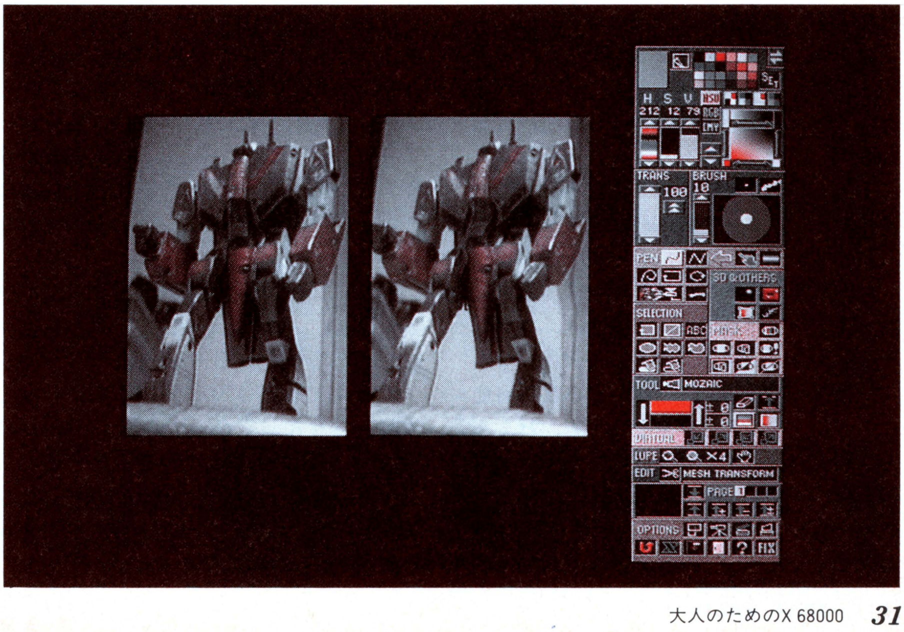


図8　表題「裸のバルキリー」



ゲーム画面保存プログラム

SAVESC.SYS

Sugimoto Riki 杉本 利貴

いつでも好きな画面をディスクにセーブして再現する画面保存システムです。
特殊な画面構成のゲームに対してもかなりの再現性を持っています。もちろんゲーム以外のプログラムにも使用できます。

ゲーム画面保存プログラムというと「5月号の質問箱SPECIALのを入力したからいいや」と思う人がいるかもしれません。それではちょっと悲しいので，最初に本プログラムの特徴を紹介します。かなり“使えるプログラム”になっているのがわかると思います。

影山氏の名誉のためにひと言申し上げますと，先のプログラムは“グラフィックパレットの保存”という質問に答えたものであり，限られたスペースでの回答であることを考え合わせると，“実現方法の一端を示し，完成は読者の努力に委ねた”のだと思います。つまり，実用的なプログラムではないが，回答としては成り立っているといえます。そのことは文章から十分にうかがえます。それでは説明に参りましょう。

特徴

1)　ほとんどすべてのゲーム（でなくてもよいが）に対応している

オリジナルのシステム上で走っていようが，パワースイッチオフのベクタを書き換えていようが平気です。垂直帰線期間，指定ラスタ，FM音源その他の割り込みを使用していても問題ありません。

2)　TEXT，GRAPHIC，SPRITE(BACKGROUND)画面，パレットをすべて保存

まあ，当然でしょうが。ちなみに5月号の質問箱のリストでは，65536色モードにしか対応していませんが，ゲーム画面の保存という意味ではちょっと問題があります。

3)　半透明，特殊プライオリティ，スクロール，特殊画面モードなどにも対応

画面制御関連のレジスタで読み出し可能なものはすべてセーブしますので，ほとんどは再現可能です。CRTとのタイミング制御(CRTC R0-R8)，TEXT/GRAPHICスクロールレジスタ（CRTC R10-R19)は読み出せませんが，別プログラム(LOADSC.X)によって対応します。

4)　画面の保存後ゲームなどに復帰できる

電源を落とさず，割り込みなどにも注意しているので可能です。とりあえず私の持っているゲームではすべて復帰できます(Yet Another Columnには一度ハマッた。プログラムを大幅に書き換えることになってしまいました)。

これによってゲームを終了することなく複数のシーンをいくつでも（フロッピーと時間のある限り）セーブでき，ポーズのきかないゲームではその代役も果たせます。(当然この場合フロッピーは必要ない)。

また，キー操作により電源を切ることもできます。

さて，次にプログラムの内容について解説します。ここを読まなくてもとりあえず使えますので，興味のない人は，“作成方法”に飛んでしまってもかまいません。

プログラムの概要

まずは，どうすればゲーム中の画面を保存できるのかを考えてみます。

画面セーブを始めるきっかけはどうすればよいでしょう。ゲームではまず使用しないものを選ばなければいけません。たとえば，ジョイスティック，マウス，キーボードの割り込みなどが適さないのは明らかです。ここでは，“質問箱SPECIAL”同様にフロントスイッチの割り込みを使用することにしました。

しかし，まだ問題は山積みです。1つひとつ解説していきましょう。

1)　オリジナルのOSにも対応する

ゲームはAUTOEXEC.BATで始まるとは限りません。それどころかディスクのIPLからオリジナルのことさえあります。こういったゲームではゲームディスクから立ち上げたが最後，ユーザーのプログラムを実行させる隙はありません。

また，多くのゲームはほかのプログラムが入り込むことを予想していないので（もしくは許していない)，メモリの確保などを自分勝手に(?)している場合があります。つまり，通常の方法で割り込みルーチンを常駐させてもあとからそのメモリを書き換えられてしまう可能性が高いのです。もし書き換えられたら，フロントスイッチオフにした途端なにが始まるか見当もつきません。暴走は序の口，悪くすればいきなりディスクを破壊してしまう可能性だってあります。

ではどうすればよいのでしょうか？　ゲームのIPLよりも先に実行され，なおかつゲームの使用できないメモリ領域に処理部を置けばよいのです。そんなことができるんでしょうか？　X68000ならできます。答えはSRAM起動です。

SRAMから起動（詳細はHuman68kユーザーズマニュアル SWITCH参照）したあとrtsすれば，BOOT＝STANDARDの順番で起動デバイスを探してくれますので，プログラム実行の順序はこれでOKでしょう。

ゲームの使用できないメモリ云々に関しては，プログラムで使用するメモリ領域をすべてSRAM上に割り当てることにします。いくら行儀の悪いプログラムでもSRAMをユーザーに断りもなく使用することはしないでしょう。どんな大事なプログラムやデータがSRAMに保存されているかもしれないですからね（そんなプログラムは認めないし，もしあったら怖くて使えない。だってほかでもなにをやっているかわからないでしょう?)。

ここまでの説明でおわかりでしょうが，本プログラムは通常呼ぶところの常駐プログラムではありません。つまり常駐させなくとも(_KEEPPRしなくてもの意味)常にそこにあるということです。

2)　パワーオフベクタを変更するプログラムに対応

ゲームにパワーオフベクタを変更されるとなぜ困るのでしょうか？　自分がせっかく書き換えたのにそれを無効にされるからです。それではこちらがあとに書き換えればいい(後手必勝)。ということで，この場合はディスク挿入・イジェクト割り込みのベクタを書き換え，その割り込みルーティンでパワーオフベクタを変更します。

3)　ゲームへの復帰ができると便利

シューティングゲームで全ボスの画面が欲しいなんてときに，一度画面保存すると電源が落ちてしまうプログラムでは，メチャクチャ効率が悪くなります。反射神経にも根気にも欠ける私には，1週間は必要でしょう。だから，これは必須です(強引か？)。1)にあるようにメインメモリを使わないようにしたり，4)で説明するようにMFPの割り込みマスクを利用したりゲームの動作環境を保存し，rteすることでゲームに復帰します。

4)　割り込みルーティンの処理

シューティングやアクションゲームにはよくあることですが，垂直帰線期間（グラフィックなどの書き換え），CRTの指定ラスタ（ラスタスクロール），FM音源（Yet Another Columun）などの割り込みルーチンで動作するものがあります。これらを止めないと画面を保存している最中に画面がどんどん変わっていってしまうとか，ゲームが止まらないなど困ったことになってしまいます。

プログラム中では，ディスク関係の割り込みは当然必要になってきますし，都合の悪いことにディイスク関係の割り込みレベルは最低位です。したがって，CPUを割り込み禁止にすることはできません。そこで，MFPの割り込みマスクを使って発生させたくない割り込みは保留状態にしておきます。これなら画面のセーブもゲームへの復帰も果たせます。また，ついでにフロントスイッチの割り込みも禁止しておくことによって，プログラムが二重に実行されるのも防ぐことができます。

5)　使いやすさも大事

本プログラムを使用するにはフォーマットずみのブランク2HDディスクが必要です。しかし，ユーザーがディスクをフォーマットし忘れているかもしれませんし，確かもう1枚あったはずなんだけど……なんてことも当然予想されます。ゲームはともかく，たとえばSX-WINDOWで作業中に実行した場合はどうなるでしょう。必要なファイルを保存していないのだが，ディスクがないために先に進めずリセットしかない……なんていうのは困ります。

そこでこのようなときには，ESCキーで復帰できるようにしました。この機能をつけ加えたために，ポーズのきかないゲームでポーズをかける（ディスクを要求してきても放っておき，解除するときにESC）なんていう技も使えるようになりました。

＊　＊　＊

さて，だいたいのところを把握してもらったので，リストを見てもらいましょう。

プログラムの説明

まず，SCREEN.EQUは画面制御関係の定数定義です。説明の必要はないでしょう。次にSAVESC.Sを見てください。これがメインルーチンでベクタの書き換えを行う部分です。1，2行目でincludeファイルを指定しています。たいていのプログラムでは，DOSCALL.MACも指定しますが，ここでは違います。Human68k上で動いていないゲームに対応するためには，当然ながらDOSの機能が使えないからです（ファイル操作が面倒だがしかたない)。

また，通常の常駐プログラムは，常駐時に余分なメモリを解放するために常駐部から書き始めるのが普通ですが，今回はメモリの解放を行わないため（メモリにプログラムを読み込んで実行するのではなく，メモリ上に常にあるプログラムに制御を移すため)，素直に実行開始部から書かれています（別にこうしなければいけないということではない)。

さて，実行される順に見ていきましょう。とりあえずレジスタを保存して，タイトルの最初の部分（共通部）を表示します。起動時に，RollUp，RollDown，Undo，カーソルキーのいずれも押されていなければ，割り込みベクタを変更せずに終了処理(skpjob：46行）に飛びます（つまり普通に立ち上げればなにもしない)。押されていたのがカーソルキーであれば，パワースイッチOFFのベクタを書き換えて終了処理にいきます。RollUp，RollDown，Undoのいずれかのキーなら2HDディスク挿入/イジェクト割り込みのベクタを書き換えます。このベクタにはdskint(63行)のアドレスが書き込まれます。

割り込みベクタを書き換えるところ（34行，83行）では，元のベクタを保存する際，$E8E00Dのアドレスに$31の値を書き込んでから保存，そのあと同アドレスに値0を書き込んでいます。SRAMはX68000の各種設定値を保存しているため不用意に書き換えられないようにこの手順を踏まなくてはなりません。

また，dskint（63行）では，ドライブ0，1にほぼ同時にディスク割り込みがかかった場合などを考慮して二重に実行されないように細工しています。また，ゲームによっては，ディスク交換が頻繁に起こるものもあるので，一度このルーチンを通過したら次回からは，オリジナルのディスク挿入/イジェクト割り込みルーチンに処理をまかせます（だからこの書き換え時には，旧ベクタをワークに保存する必要はない)。

さて，プログラム終了処理ですが，これは簡単で各レジスタの値を復帰したあとに

SRAMディスクについて

通常SRAMディスクを使用していた人ならよいのですが，人によってはドライブ数が増えることが望ましくない場合もあります。こういった場合には，SRAMDISK.SYSをはずすこともできます。

SWITCH.Xで見たときにSRAMディスクが設定されていなくてもSRAMの内容は保持されますので，いったんSRAMDISK.SYSを登録しファイルを転送したあと，SWITCH.XでSRAMの使用状況を変更してください。これでSRAMディスクが設定されていなくてもSAVESC.SYSから起動できるようになります。ただし，これはメーカー推奨の使い方ではありませんので注意してください。

また，付録ディスクで何度か使用したことのあるDRV.R(RAMディスクのドライブ名を探すプログラム）ではAドライブから順にRAMディスクを探しますので，通常のRAMディスクよりも前にSRAMディスクが設定されているとSRAMディスクのドライブ名を返してきます。

これに気づかず，そのままDRV.Rで得たドライブ名をテンポラリパスとして指定すると，テンポラリファイルを作成するプログラムのほとんどが実行できなくなってしまうことがありえます。

これを避けるため，DRV.Rを使っている人は，必ずRAMDISK.SYSよりもあとにSRAMDISK.SYSを登録するようにしてください。

また，SAVESC.SYSを使用していて起動がうまくいかないことがある場合（特にX68000PRO系列)，SRAMディスクに入ったSAVESC.SYSの内容と元のSAVESC.SYSの内容が一致しているかどうかを，

　FC /B A:SAVESC.SYS E:SAVESC.SYS

などのようにして確認してみてください。あまりに頻繁にSRAMの内容が破壊されるようでしたら，SAVESC.SYSの使用を中止しシャープに問い合わせてみてください。

そのほか，SRAM容量とプログラムの性格上，ウイルスチェックプログラムのDOCTOR2.Rなどと，このプログラムを同時に使用することはできません。

rtsすれば，SWITCH.Xで，BOOT＝STANDARDを指定したのと同じ状態でシステムを起動できます(詳細は，Human68kユーザーズマニュアルを参照)。

rtsの前にはキーバッファをクリアしなければなりません（k_flsh：55行)，まずキーが離されていることを確認してから(wait：48行）行わないとプログラムを抜けたあとも押し続けている状態になることがあるので注意が必要です（キーリピートよりキーバッファクリアの処理が高速な場合)。

この確認は，IOCSコール_BITSNSで行っていますが，その前に空ループ（時間稼ぎ）があります。これを削除するとキーを押し続けていても先に進んでしまいます(k_flshに遅延ループを入れてもよい)。DRAM上でテストすると大丈夫なのですが，SRAM上だとこのようになることを確認しました（SRAMのほうが処理が高速なためだろうか？　あまり自信はないが)。

次は，SAVESC_SUB.Sです。これは，パワースイッチOFF割り込みで実行される部分です。処理後は，ゲームが続行できるようにしたいのでワーク，バッファなどはすべてSRAM上の固定領域に取ります（すべてデータセクションもしくはテキストセクションに置くということ)。したがってSRAMに頻繁に書き込みを行うことになるので最初にSRAMを書き込み許可にしておきます（SAVESC.Sを参照)。

また，2HDディスク，キーボード関連の割り込みを使用するためCPUを割り込み許可状態にしておかなければなりません(割り込みルーティンに制御が移った時点でSR(Status Register)は割り込み禁止状態になっている)。ただし，単純に割り込みを許すとゲーム中に垂直帰線期間，CRTCの指定ラスタやFM音源割り込みで動作するルーチンがあった場合に，たとえば背景のスクロールが止まらないなどの不都合が起きてしまいます（例：沙羅曼陀，NAIOUS, Yet Another Columnなど)。

そこで，MFPに細工します。このLSIは割り込みを制御しているものです(RS-232C，マウス，ディスクドライブなどは除く)。つまり，MFPで必要のない割り込みを禁止しておきCPUを割り込み許可にすることで，不都合な割り込みを無効にすることができるのです（sv_mfp：347行)。

まず，電源を切ろうとする割り込みを無効にします。これで，本プログラム実行中にさらに同じプログラムが動くのを防げます（350行は，andi.b #%1111_1011,(a0)でもよいが，まあついでということで)。

次に，MFPの制御する割り込みをキーボード関連を除きすべてマスクします。このマスクというのは，禁止するのと違ってとりあえず割り込みを保留しておき，あとでその割り込みを行わせることができるのです。したがって，音楽等を止めたあと（こうしておかないとYet Another columnなどは止まらない）再び演奏を再開させることも可能となります。また，ラスタスクロールに関しても同じことがいえます。

さあ，CPUの割り込みを許可すれば（16行）これで用意はできました。まず，ユーザーが指定のキーを押していないかどうか調べます。起動時に割り込みベクタを変更させることを指示するキー（押したのと同じでなくてもよい）が押されていたなら本来のパワースイッチOFFのルーチンに飛びます（off：339行)。内容は各種レジスタの復帰，SRAM書き込み禁止のあと，割り込みベクタ保存時に取っておいたアドレスをスタックに積みrtsします。

キーを押していないならいよいよ画面を保存することにします。細かい解説の前に大まかに動作の流れを説明します。パワースイッチが切られ，なおかつ指定キーを押していないなら，ドライブ１にディスクがあればイジェクトしないときはそのままLEDを点滅させ，挿入を促します。この際，うっかりディスクを用意し忘れたなどのミスを考慮しESCキーが押されたら画面セーブを中止しゲームなどに戻ります。

無事にディスク挿入を確認したら現在の画面をHuman68kで扱える形式のファイルとして保存しディスクをイジェクトします。最初にディスクが入っていた場合は，再び挿入を待ち，入っていなかった場合はそのままゲームに戻ります。

この間，たとえばディスクのエラー（使用ずみディスク，ライトプロテクトなど）があっても一切メッセージは出しません。もちろん，テキスト画面の一部待避，プライオリティの待避と設定，テキストスクロールレジスタの設定，場合によっては画面モードの待避と変更を行えばメッセージを出すことも可能です。ただし，SRAM上ですべて賄うためにプログラム領域，バッファなどが肥大します。貴重なSRAMを喰いつぶすのは得策ではないと判断し断念しました（面倒臭いのがいちばんの理由だったりもするが)。

さて，プログラムそのものはさほど難しいこともないし，コメントも詳しく（普段はこの1/10程度しか書かない）書いたつもりなので，ポイントだけ解説していきます。

ここで使用するディスクは，フォーマットずみでなにもファイルの入っていないものに限ります（ディスクの使用可能セクタが飛び飛びにある場合などDOSに頼らず処理するのが困難なため)。49行以下でルートディレクトリを調べ，なにもファイルがないことを確認します（消去されたファイルは許す)。

ディスクの正当性を確認したら，いよいよデータを書き込んでいきます。この際FATもワーク上に作成していきますが，2HDディスクの1.5バイトFATは複雑なためとりあえず２バイトFATで作成し，あとでまとめて1.5バイトFATに変換するとい

SCREEN.EQU

今回紹介したSAVESC.SYSとLOADSC.Xは，プログラム中で参照しているX68000の画面管理レジスタやG-RAM，スプライトPCGエリアなどのアドレス情報を“SCREEN.EQU”として分離し，外部定義シンボルとして参照するようになっています。

ですから，ソースリストからプログラムを打ち込み，アセンブルしてSAVESC.SYSを作成する際にはこのファイルをカレントディレクトリに置いておく必要があります。しかし，掲載スペースが中途半端だったので，SCREEN.EQUはその他のリストと違って後ろにまとめて掲載することができませんでしたので，SCREEN.EQUはここに掲載しておきます。

ソースリストからプログラムを作成する場合には，右の小さなリストをED.Xなどのテキストエディタから入力し，“SCREEN.EQU”の名前でセーブしておいてください。

なお，ダンプリストを打ち込んで実行ファイルを取り出す場合にはこのリストは必要ありません。

```
*
*    SCREEN.EQU
*
G_VRAM    equ $C00000
T_VRAM    equ $E00000
G_PAL     equ $E82000
T_PAL     equ $E82200

SP_REG    equ $EB0000
BG_REG    equ $EB0800
SP_MOD    equ $EB080A
PCG       equ $EB8000

CRTC_20   equ $E80028
VCTRL_0   equ $E82400
VCTRL_1   equ $E82500
VCTRL_2   equ $E82600

TX_SCR    equ $E80014
```

う方法を取りました。

このファイルに保存されるデータとしては,

CRTC(CRT CONTROLLER)　R20
VCTRL(VIDEO CONTROLLER) R0, R1, R2
TEXT VRAM
GRAPHIC VRAM
GRAPHIC PALETT
TEXT PALETT
SPRITE PALETT
SPRITE / BACKGROUND ALL SORTS OF REGISTERS

があります。しかし，実は画面を完全に復元するにはこれだけでは不足です。あと必要なのは，CRTC R00～R08，R10～R19となります。

これらのレジスタは，書き込みのみで読み出しができないため保存することは不可能です。このうちR00～R08はCRTとのタイミングをコントロールするためのもので，画面モードがわかれば標準の値を設定することができます。したがってここではとりあえず無視してファイルを画面にロードするなどの処理時に対応することにします。

残りのR10～R19は，テキスト，グラフィック画面のスクロールレジスタです。これらの値で各画面の表示位置が決まるため正しい値が設定されていないと背景とキャラクタの関係がおかしくなったりすることになります。まあ，読み出せないものはしかたないので，とりあえずスクロールレジスタのデータ用に場所を確保して0で初期化しておきます。こうしておけば，あとで別プログラムによって設定し保存することが容易になるからです。

データのセーブ形式は，ほとんどメモリ上にあるまま（未圧縮）です。グラフィックVRAMに関しては，有効ビットのみをセーブします(65536色モード＝16ビット，256色モード＝８ビット，16色モード＝４ビット：いずれも512Kバイトのデータとなる)。

種々のデータをディスクに書き込んだら，FATデータを元にルートディレクトリ用のデータを作成し，Human68kのファイルとして扱えるようにします。このあたりは，コメントにしつこく書いてあるためリストを見れば理解できるでしょう。ここでも，エラーメッセージを出さないのと同様の理由で(括弧内まで同じである)，日時の取得は省略しています。

画面制御に関係するプログラムを作ったことのある人でしたら，その他のところは容易に理解できると思います。よくわからないという人は，誌面の都合もありますので後述の参考文献を調べるか，Oh!Xのバックナンバーで「X68000 マシン語プログラミング」(村田敏幸氏)，「吾輩はX68000である」(泉大介氏)などを，読み返してください。

あ，それからいつもアセンブラでプログラムしている方も次の「作成方法」は読んでください。ここに書かれているとおりにしないと動きませんので。

作成方法

SAVESC.SYSとLOADSC.Xの２本のプログラムは圧縮されたものをダンプリストで掲載します。６月号付録ディスクのなかからMAC.Xを取り出して入力してください。セーブ時のファイルサイズは2850バイトを指定してください。これをLHA.Xで展開すれば実行ファイルが得られます。

次にソースリストから実行ファイルを作成する際の注意をまとめます。まず，SCREEN.EQU，SAVESC.S，SAVESC_SUB.Sの３つをエディタ（ED.Xなど）で入力してください（行番号は入力してはいけない)。念のためディスクにセーブしたら，カレントディレクトリにSCREEN.EQUがある状態で，

AS SAVESC
AS SAVESC_SUB

を実行してください（AS.Xを使用する場合。HAS.Xでもよい)。SAVESCはなにもWARNINGは出ません。SAVESC_SUBでは，Absolute addressingが７つ出る状態でOKです(結果が違うときは，先に進まないこと)。

SAVESC.O, SAVESC_SUB.Oができたら，

LK /B ED0C40 SAVESC SAVESC_SUB

を実行してください（HLK.Xは不可。/Bができない)。/B ED0C40は，実行アドレスの指定なので間違いのないように気をつけてください。

無事にSAVESC.Xができたなら，誤って実行しないように（コマンドラインから実行してはいけない）SAVESC.SYS にリネームしておきます。

REN SAVESC.X SAVESC.SYS

次に，SRAMDISK をフォーマットしてからこれをコピーします（SRAMDISKを使っていない人は，Human68kユーザーズマニュアルを参照して作成してください)。SRAMDISKを使わずにSRAMをプログラム領域として使用する方法もありますのでそちらを採用してもかまいません。ただし，リンク時のアドレス指定とSWITCH.Xで設定するブートアドレスは変更する必要があります。私は，SRAMDISKも使いたいのでこの方法を取っています。

FORMAT ?:
COPY SAVESC.SYS ?:
?(A-Z)はシステム構成によって異なる

SRAMは，なにかの拍子に内容が失われること（立ち上げ時にシフトキーをつい押してしまったとか）があるので，フロッピーディスクなどにも保存しておくことをおすすめします。

最後に,SWITCH.Xで起動デバイスを切り替えます。

SWITCH

と入力したら，BOOTの設定を“RAM1 $ED0C40”にしてください（Human68kユーザーズマニュアル参照)。

これで，リセット後よりSAVESC.SYSが使えるようになります。

使用方法

作成方法で述べたようにシステムをセットした状態で行ってください。

●SAVESC.SYSを使わないとき

普通はなにも押さずに立ち上げます。ゲームなどのおまけ機能でなにかを押しながら起動する必要があったときなどは，一覧表に示すキー以外なら押すことができます(なるべく,シフトキーは押さないほうがよいでしょう。SRAMが初期化されてしまいます。やむをえず行う場合はバックアップをとってください)。万一，表中のキーと一致してしまった場合は，SAVESC.SYSを起動しても通常終了する方法がありますので，参照してください。

ただし，本プログラムは起動したときのキーが離されるのを待ってシステムを起動しますので，起動メッセージが出たら一度キーを離し，すぐさまもう一度キーを押すという方法を取ってください。

●SAVESC.SYSを使うとき

まず，フォーマットずみで使用していない2HDディスク（新品でなくてもかまわないがファイルがあっ てはならない）を保存したい画面の枚数分用意してください。

これから使おうとするソフトが，フロントスイッチを切るときPCMを鳴らしたり，電源を切ってもよいか尋ねてきたりする場合は，一覧表の“パワーオフベクタが書き

換えられている場合”のキーを押しながら立ち上げます（電源を入れるかリセットする)。それ以外のときは“通常”のキーです。また，“通常”のキーで立ち上げて，SAVESC.SYSが機能しない場合は，もう一度“パワーオフ”立ち上げの方法で試してみてください。
“通常”立ち上げの場合はこれで準備完了ですが，“パワーオフ”立ち上げの場合は，ゲームなどが立ち上がったあとに，一度ディスクを挿入もしくはイジェクトしてください（ゲームがベクタを書き換えたあとに実行するということ。２ドライブ使用のソフトなら本来のディスクをもう一度挿入するのを忘れないように)。

さて，「この画面を保存したい！」というときがきたら，すかさずフロントスイッチを切ります。ドライブ１がディスク挿入を促しますので，用意した2HDディスクをセットします。ESCキーを押せばゲームに戻れます（これを利用してポーズ機能としても使える)。用意周到なあなたには，ディスクアクセスが始まりますのでしばらく待ちます。

そしてデータセーブがすむと，フロントスイッチを切ったときにドライブ１にディスクが入っていなかった場合は，すぐさまゲームなどに復帰します。入っていたときは元のディスクが挿入されるのを待って復帰します。この際フロントスイッチはOFFのままでもX68000は走り続けます。このままでもかまわないのですが，本プログラムが走っているあいだにスイッチを入れ直すことをすすめます。理由を説明しますと，フロントスイッチを入れ直した場合に割り込みがかかることがあるからです。必ずかかるのならそれはそれでよいのですが，そうでないこともあるというのが困りものなんです（想像してみればわかると思います)。本プログラム中では，この割り込みを無効にしていますので，実行中は心配はないというわけです。

このプログラム実行は，フロッピーのある限り何回でも可能です。つまり，複数の画面セーブを一度に（立ち上げ直さずにという意味）できるのです。

さて，やがて本当に電源を切りたくなるときがやってきます。そのときは，一覧表中のどれかひとつのキーを押しながら電源を切ってください。また，SAVESC.SYSを使いたくないのに，ゲームのおまけなどで押すキーと一致していたために本プログラムが働いてしまっているときも同様にしてください（フロントスイッチOFFの状態でリセットし直してもいいのだが，カッコ悪いのでこうした)。

とりあえず表示

さて，ディスクにセーブしただけでは，面白くありません。保存した画面を表示してみましょう。先ほど保存できなかったテキスト/グラフィック画面の表示位置や，CRTとのタイミングを設定するレジスタもここで対応します。LOADSC.Xです。主な処理内容はディスクに書き込まれたデータをX68000に戻していくだけなのでSAVESC.SYSを理解できればほとんど説明の必要はないでしょう。

ここで，CRTCのR00～R08までに設定しているデータは，あくまで標準的な値にすぎないので，特殊な値を設定しているソフトなどには対応していません。というよりも，汎用的なプログラムとしては対応できないので，もしそのような画面を表示したい場合は個別に設定しなければならないでしょう。また，ラスタスクロールなどにも対応していません（スクロールしていないそれなりの絵は表示できる)。

それでは表示してみましょう。書式は，

LOADSC ［/S］ FILE NAME

となります。オプションスイッチは，/，-，S，sのどれでも使えます。位置もファイルネームの前後どちらでもかまいません。

/? などの無効なオプションを指定すると使用法を表示します。また，ファイル名省略時はカレントディレクトリよりSCREEN.DATを探します。これはドライブ，ディレクトリのみの指定でも同じです。

/Sが指定された場合は，画面表示の後表示位置設定モードになります。ディスクアクセスが終了したら，カーソルキー（矢印キー）を押してみてください。テキスト画面になにか表示されていれば，それがスクロールします。スペースキーを押し，またカーソルキーを押すと今度はグラフィック画面のページ０がスクロールします。スペースを押すたびに次の画面に移っていき，グラフィック画面の最終ページ（画面モードによって違う）を設定したあと，再びテキスト画面に戻ってきます。

スピードはさほどないので１ドット単位の修正も容易だと思います。XVIを持っている人には，速すぎるかもしれないので，OPT.1キーと同時に押すとスピードダウンするようにしてあります（私はACEユーザーなのでよくわからない)。

さて，すべての画面表示位置を設定したら，“S”のキーを押してください。現在の画面表示位置がディスクに書き込まれて，これ以後は設定された位置に表示されるようになります（もちろんあとから変更することも可能)。

すべての作業が終了し，ESCキーを押せば終了します。Sを押さずにESCを押せば，いま調整した表示位置を書き込まずに終了することもできます。

/Sが指定されていない場合は，単に表示して終了します。

さて，このプログラムではいろいろな使用状況に対応するために，ファンクションキーやカーソルの消去を行っていません。おまけとして，これらを消すバッチファイルCON2.BATを付けておきます（電脳倶楽部で発表されたCONCON.Rを意識した名前である)。使用法は，CONCON.Rと同様で，CON2 /?などでヘルプも出ますので使ってみてください。

画面を観賞(?)したあと，SCREEN命令で消去してみてください。全部消えましたか？　テキスト画面のプレーン２・３（普段マウスカーソルやソフトキーボードが表示されている）になにか書き込まれていたらそれは消えません。そこで，おまけその２はCLRALL.Rです。テキスト画面をすべて消去します。

さらなる発展

さて，画面を保存し，好きなときにそれを見られるようになりました。が，これで終わりではありません（今回は終わりだが)。最大の問題は作成されるファイルのサイズが大きすぎることでしょう。そもそも，このファイルはこのまま扱うことを想定していません。とりあえずメモリの内容を保存して，Human68k上のプログラムで加工することを目的としています。効率・手間を考えると当然の判断であると思います。

また，データをもっと有効に使うことも検討するべきです（たとえば，ほかのソフトに取り込みが可能なかたちにするとか)。私も現在いくつかのアイデアを持っておりますが，皆さんもなにかユニークな活用を考えてみてはいかがでしょうか。

開発後記

“ゲーム画面の保存”ということで説明してきましたが，もちろんゲーム以外にたとえばSX-WINDOW，VS.X，グラフィックツールなどにも使用できます（要するにな

んでもあり)。しかし，ゲームなどはなにをやっているかわからない（ディスクで立ち上げ，電源を切ることで終了することを前提としているので，ほかのプログラムのことは考慮していない）ので対応できないものもあるかもしれません。たとえばSAVESC.SYSが使えないようなゲームをわざと作ることは難しいことではありません（でも，まず大丈夫だと思いますが)。

書体倶楽部のアウトラインフォントで文字を表示しておきディスクに保存，それをMultiWordのグラフィックデータにコンバート，ワープロにアウトラインフォントが使える……ゲーム画面をSX-WINDOWのPIX形式にする……なんて応用も容易にできます(もちろん多少の知識は必要)。まあ，これを使ってゲームを作り，市販して……というのは，著作権を考えると無理ですが，個人で楽しむ分には問題はないでしょう。この原稿を書き終わったらいろいろ作ってみるつもりです。

それから，ゲーム画面を再表示するというのは，意外と勉強になるものです。たとえば，ゲームをやっているときにはスプライトで表示していると思っていたものがグラフィック画面だったり，テキスト画面だと思っていたのがスプライトだったりなんてこともあります。　また，スプライトの表示/非表示を高速に繰り返し半透明のような効果を出していたり，特殊プライオリティを実にうまく使っていると感心させられるものもありました。あ，一応このあたりを判断できるように，LOADSC.Xが表示していく順番を書いておきましょう。

1．TEXT画面

プレーン０～３の順にディスクから直接書き込む。

2．GRAPHIC画面

まとめて読み込みまとめて表示（メモリ容量による)。

3．PALETT　(GRAPHIC，TEXT，SPRITE)

4．SPRITE，BACK GROUND

＊　＊　＊

今回のプログラムは，2HDディスクへの直接的な書き込み，MFP，CRTC，SRAM起動など，私にとって初めての経験がたくさんありました。以前に実験（イタズラ）程度にいじったことはあったものの完成されたプログラムにするには，“ハマリ”が多く苦労しました。しかし，私は趣味のプログラマなので“苦労”というよりも“楽しみ”が多かったという気もします。完成できたのはInside X68000に負うところが大きく，桑野氏には大変感謝しております(う～ん，ちょっと間違いも見つけたけど)。

さて，次回はどんなプログラムでお目にかかれるかお楽しみに。

参考文献

1)　村田敏幸，「X68000マシン語プログラミング」，Oh!X，ソフトバンク

2)　桑野雅彦，「Inside X68000」，ソフトバンク

3)　吉沢正敏・市原昌文，「X68000環境ハンドブック」，工学社

4)　千葉憲昭，「X68000ベストプログラミング入門」，技術評論社

リスト1. SAVESC. S

```
        .include        iocscall.mac

CR      equ     $0D
LF      equ     $0A

        .xref   savesc
        .xdef   lstvct

        .text
        .even

        bra     ent                     *最初はこの命令でなければならない
ent:    movem.l d0-d1/a0-a1,-(sp)
        movea.l #$E8E00D,a0
        move.b  #$31,(a0)               *SRAM書き込み許可
        clr.b   intflg
        clr.b   (a0)
        lea.l   title1(pc),a1
        IOCS    _B_PRINT                *TITLE 途中まで表示

        move.w  #7,d1
        IOCS    _BITSNS                 *RollUp, RollDown, Undo, Cursol key
        andi.b  #%111_1111,d0           *何も押してなければ何もしない
        beq     skpjob

        move.b  d0,d1
        lea.l   title2(pc),a1
        IOCS    _B_PRINT
        andi.b  #%111,d1                *Cursol Key なら
        beq     skpdsk                  フロントスイッチのみ書き換え
        lea.l   title4(pc),a1
        IOCS    _B_PRINT

        move.w  #$61,d1                 *2HD DISK の
        lea.l   dskint,a1               *挿入・イジェクト割り込みで
        IOCS    _B_INTVCS               *フロントスイッチの割り込み
        lea.l   dskvct(pc),a1           *先を書き換える
        move.b  #$31,(a0)
        move.l  d0,(a1)
        clr.b   (a0)
        bra     wait

skpdsk: bsr     sw_vct
        bra     wait

skpjob: lea.l   title3(pc),a1
        IOCS    _B_PRINT
        bra        return
wait:   moveq.l #$40,d1                 *起動時のキーが離されるのを待つ
wait0:  dbra    d1,wait0
        moveq.l #7,d1
        IOCS    _BITSNS
        tst.b   d0
        bne     wait
k_flsh: IOCS    _B_KEYSNS               *キーバッファのクリア
        tst.w   d0
        beq     return
        IOCS    _B_KEYINP
        bra     k_flsh
return: movem.l (sp)+,d0-d1/a0-a1
        rts                             *STANDARD の順序で起動する
*----------------------------------------
dskint: movem.l d0-d1/a0-a2,-(sp)
        lea.l   intflg(pc),a2
        tst.b   (a2)                            *現在処理中なら戻る
        bne     dskrtn
        movea.l #$E8E00D,a0
        move.b  #$31,(a0)                       *SRAM 書き込み許可
        st.b    (a2)                            *処理中であることをしめすフラグ
        clr.b   (a0)
        bsr     sw_vct                          *フロントスイッチ処理の書き換え
ld_dvct:
        move.w  #$61,d1                         *一回でこの処理は用済みなので
        movea.l dskvct(pc),a1                   *元に戻しておく
        IOCS    _B_INTVCS
        move.b  #$31,(a0)
        clr.b   (a2)
        clr.b   (a0)
dskrtn: movem.l (sp)+,d0-d1/a0-a2
        move.l  dskvct(pc),-(sp)
        rts
*----------------------------------------
sw_vct: move.w  #$42,d1
        lea.l   savesc(pc),a1
        IOCS    _B_INTVCS
        lea.l   lstvct(pc),a1
        move.b  #$31,(a0)               *SRAM 書き込み許可
        move.l  d0,(a1)
        clr.b   (a0)                    *SRAM 書き込み禁止
        rts
*----------------------------------------
        .data
        .even
lstvct: .dc.l   0
dskvct: .dc.l   0
intflg: .dc.b   0
title1: .dc.b   '1mSAVESC ver1.00 1992 Ricky Sugimoto1m',CR,LF
        .dc.b   'スクリーン(Graphic/Text/Sprite/Back Ground)保存システム登録',0
title2: .dc.b   'します',CR,LF,0
title3: .dc.b   'しません',CR,LF,0
title4: .dc.b   'ディスクの挿入/イジェクト割り込みによってベクタを書き換えます',CR,LF,0
        .end
```

リスト2. SAVESC SUB. S

```
        .include        iocscall.mac
        .include        screen.equ

        .text
        .even

        .xdef   savesc
        .xref   lstvct

savesc: move.b  #$31,$E8E00D            *SRAM 書き込み許可
        move.l  sp,spsave
        lea.l   stackbtm(pc),sp
        movem.l d0-d7/a0-a2,-(sp)

        bsr     sv_mfp
        andi.w  #$F8FF,sr               *割り込み許可

        move.w  #7,d1                   *RollUp, RollDown, Undo
        IOCS    _BITSNS                 *Cursol key のどれかが
        andi.b  #%0111_1111,d0          *押されていれば電源を切る
        bne     off

        move.w  #$9100,d1
        moveq.l #0,d2
        IOCS    _B_DRVCHK
        move.w  d0,dskmd                *現在の drive 0 の状態を保存
        moveq.l #3,d2                   *イジェクト許可
        IOCS    _B_DRVCHK
```

```
         moveq.l #4,d2              *LED 点滅
         IOCS    _B_DRVCHK

baddsk:  bsr     b_eject            *Disk Eject

insert:  moveq.l #0,d1              *ESC key
         IOCS    _BITSNS            *を押していたら終了処理へ
         btst.l  #1,d0
         bne     exit
         move.w  #$9100,d1
         IOCS    _B_DRVCHK
         btst.l  #1,d0
         beq     insert             *Disk が挿入されるのを待つ
         btst.l  #3,d0              *Write Protect Sheel
         bne     baddsk             *が貼られていたら EJECT
         btst.l  #0,d0              *誤挿入を EJECT
         bne     baddsk
         btst.l  #2,d0              *Drive の準備ができていなければ待つ
         bne     insert

         move.w  #$9170,d1          *Drive 0 の
         move.l  #$0300_0006,d2     *Root Directory を
         move.l  #1024*6,d3         *buff 以下へ読み込む
         lea.l   buff(pc),a1
         IOCS    _B_READ

         lea.l   buff(pc),a0
         move.w  #1024*6/32-1,d6
rdirck:  tst.l   (a0)               *0 なら
         beq     okdsk              *Root Directory の終わり
         cmpi.b  #$E5,(a0)          *$E5 なら消去されたファイル
         beq     nxtdir             *次を調べる
         bra     baddsk             *何かファイルがあるなら別のディスクを要求
nxtdir:  adda.l  #32,a0
         dbra    d6,rdirck
okdsk:   moveq.l #5,d2              *LED 消灯
         IOCS    _B_DRVCHK

         lea.l   fat2(pc),a0        *fat2, fat1_5 を使ったので
         move.w  #1024*5/4-1,d6     *念のためにクリア
clrfat:  clr.l   (a0)+
         dbra    d6,clrfat
         lea.l   fat2(pc),a2        *a2 = fat2

         lea.l   buff(pc),a0

         move.w  CRTC_20,d7         *CRTC R20, VCTRL R0 - R2
         move.w  VCTRL_2,d6         *を読み込む
         move.w  d7,(a0)+           *
         move.w  VCTRL_0,(a0)+
         move.w  VCTRL_1,(a0)+
         move.w  d6,(a0)+
         move.b  d6,d7              *d7:上位 8bit = CRTC R20, 下位 8bit = VCTR
, R2
         moveq.l #5*4-1,d6
scr_lp:  clr.w   (a0)+              *TEXT, GRAPHIC 画面のスクロールレジスタ用
         dbra    d6,scr_lp          *とりあえず0で初期化

         move.w  #$9170,d1
         move.l  #$0300_0103,d2     *d2=sector
sv_mod:  move.l  #2*4+4*5,d3
         lea.l   buff(pc),a1
         bsr     write              *SAVE CRTC R20, VCTRL R0 - R2
         tst.w   d0
         bmi     sv_mod

         movea.l #T_VRAM,a0
         move.w  #512-1,d6
sv_txt:  movea.l a0,a1
         move.l  #1024,d3
         bsr     write              *SAVE TEXT VRAM
         tst.w   d0
         bmi     sv_txt
         move.l  a1,a0
         dbra    d6,sv_txt

         move.w  d7,d0
         lsr.w   #8,d0
         andi.b  #3,d0              *d0: 0=16col, 1=256col, 3=65kcol
         tst.b   d0
         beq     col16
         btst.l  #1,d0
         beq     col25

col65k:  movea.l #G_VRAM,a0         *65,536 色モード時
         move.w  #512-1,d6          *グラフィック画面のセーブ
lp65k:   movea.l a0,a1
         move.l  #1024,d3
         bsr     write
         tst.w   d0
         bmi     lp65k
         move.l  a1,a0
         dbra    d6,lp65k
         bra     svtpal

col25:   movea.l #G_VRAM,a0         *256 色モード時
         move.w  #512-1,d6          *グラフィック画面のセーブ
c25lp0:  lea.l   buff(pc),a1
         move.w  #1024/2-1,d5
c25lp1:  move.w  (a0)+,d0
         move.w  (a0)+,d4
         lsl.w   #8,d0
         andi.w  #$FF,d4
         or.w    d4,d0
         move.w  d0,(a1)+
         dbra    d5,c25lp1
sv25:    lea.l   buff(pc),a1
         move.l  #1024,d3
         bsr     write
         tst.w   d0
         bmi     sv25
         dbra    d6,c25lp0
         bra     svtpal

col16:   movea.l #G_VRAM,a0         *16 色モード時
         move.w  #512-1,d6          *グラフィック画面のセーブ
c16lp0:  lea.l   buff(pc),a1
         move.w  #1024-1,d5
c16lp1:  move.w  (a0)+,d0
         move.w  (a0)+,d4
         lsl.b   #4,d0
         andi.b  #$F,d4
         or.b    d4,d0
         move.b  d0,(a1)+
         dbra    d5,c16lp1
sv16:    lea.l   buff(pc),a1
         move.l  #1024,d3
         bsr     write
         tst.w   d0
         bmi     sv16
         dbra    d6,c16lp0
         bra     svtpal

svtpal:  movea.l #T_PAL,a0
         lea.l   buff(pc),a1
         move.w  #512/4-1,d6        *TEXT & SPRITE PALETTE のバイト数 -1
         btst.l  #6,d7
         bne     tpallp
         move.w  #32/4-1,d6         *TEXT PALETTE のバイト数 -1
tpallp:  move.l  (a0)+,(a1)+
         dbra    d6,tpallp
         cmpa.l  #$E82300,a0
         bhi     svgpal
         adda.l  #512-32,a1         *SP PAL をセーブしていない時書き込み位置を調節

svgpal:  movea.l #G_PAL,a0
         move.w  #32/4-1,d6         *16 色モード時の GPAL Byte - 1
         btst.l  #8,d7
         beq     gpallp
         move.w  #512/4-1,d6        *256/65k 色モード時の   "
gpallp:  move.l  (a0)+,(a1)+
         dbra    d6,gpallp
         lea.l   buff(pc),a1
         move.l  #1024,d3
svpal:   bsr     write              *TEXT PALETTE, (SPRITE PALETTE)
         tst.w   d0                 *GRAPHIC PALETTE
         bmi     svpal

         btst.l  #6,d7              *SPRITE を表示していないなら
         beq     mk_fat             *FAT の作成に飛ぶ

         movea.l #SP_REG,a1
         move.l  #1024,d3
svsreg:  bsr     write              *SPRITE SCROLL REGISTER
         tst.w   d0
         bmi     svsreg

         movea.l #BG_REG,a1
         move.l  #18,d3
svgreg:  bsr     write              *BACK GROUND SCROLL REGISTER
         tst.w   d0
         bmi     svgreg

         movea.l #PCG,a0
         move.w  #32-1,d6
sv_pcg:  movea.l a0,a1
         move.l  #1024,d3
         bsr     write              *PCG AREA, (TEXT AREA)
         tst.w   d0
         bmi     sv_pcg
         movea.l a1,a0
         dbra    d6,sv_pcg

mk_fat:  move.w  #$FFF,-2(a2)       *End of FAT
         lea.l   fat2(pc),a0        *a0 = fat2
         moveq.l #3,d0              *d0 = sector number
chkbad:  cmpi.w  #$FF7,(a0)         *先頭のセクタは正常か
         bne     chknxt             *正常なら次のセクタのチェックへ
         addq.l  #1,d0              *異常ならセクタナンバを増やす
         addq.l  #2,a0              *注目しているファット位置を1つ進める
         bra     chkbad             *再びチェック
chknxt:  cmpi.w  #$FFF,(a0)         *このセクタで終わりか?
         beq     secend             *処理終了
         move.w  d0,(a0)            *取敢えず次のセクタナンバを書き込む
         addq.l  #1,d0              *セクタナンバの更新
         movea.l a0,a1              *書き込んだファット位置を覚えておく
         addq.l  #2,a0              *注目しているファット位置を1つ進める
cknxt0:  cmpi.w  #$FF7,(a0)         *使用不能か?
         bne     chknxt             *正常ならさっき書き込んだのは良かった。
         addi.w  #1,(a1)            *異常ならもう一つ増やす
         addq.l  #2,a0              *注目するファット位置を1つ進める
         addq.l  #1,d0              *セクタナンバも増やす
         bra     cknxt0             *その次をチェック
secend:                             *これで2バイトファットができた
         lea.l   fat2(pc),a0        *1．5バイトファットに直す
         lea.l   fat1_5(pc),a1
         moveq.l #0,d4
         move.w  #$FEFF,(a1)+          *Media byte
         move.b  #$FF,(a1)+            *        "
chg_15:  move.w  (a0)+,d0
         cmpi.w  #$FFF,d0
         bne     c_150
         moveq.l #-1,d4
c_150:   move.b  d0,(a1)+
         lsr.w   #8,d0
         move.b  d0,(a1)
         tst.w   d4
         bmi     c_15ed
         move.w  (a0)+,d0
         cmpi.w  #$FFF,d0
         bne     c_151
         moveq.l #-1,d4
c_151:   lsl.w   #4,d0
         or.b    d0,(a1)+
         lsr.w   #8,d0
         move.b  d0,(a1)+
         tst.w   d4
         bmi     c_15ed
         bra     chg_15
c_15ed:
         move.w  #$9170,d1
         move.l  #$03_00_00_02,d2
         move.l  #1024*2,d3
         lea.l   fat1_5(pc),a1
         IOCS    _B_WRITE

         lea.l   fat2(pc),a0        *最初に書き込んだセクタを捜す
         moveq.l #2,d0
get1st:  cmpi.w  #$FF7,(a0)
         bne     got1st
         addq.l  #2,a0
         addq.l  #1,d0
         bra     get1st
got1st:  move.w  d0,d4
         lsl.w   #8,d4
         lsr.w   #8,d0
         or.b    d0,d4
         lea.l   sec1st(pc),a0
         move.w  d4,(a0)

getsiz:  lea.l   fat2(pc),a0           *ファイルサイズを得る
         moveq.l #0,d0
         move.w  #$FF7,d4
         move.w  #$FFF,d2

gsiz0:   cmp.w   (a0),d4
         beq     skpsiz
         addq.l  #1,d0
```

▶関西にはカプコン提供の「あすの天気」という番組があります。そこで「兵庫県，はれ」といっているバックで，ストリートファイターIIがBGM付きで画面に出ているため，そっちに気をとられ肝心の天気を忘れてしまう。ありゃ反則ですよ，カプコンさん。

菅谷　英明(26)兵庫県

```
         cmp.w   (a0),d2
         beq     endsiz
skpsiz:  addq.l  #2,a0
         bra     gsiz0
endsiz:  moveq.l #10,d4
         lsl.l   d4,d0
         move.w  #4-1,d6
mksiz:   lsl.l   #8,d4
         move.b  d0,d4
         lsr.l   #8,d0
         dbra    d6,mksiz
         lea.l   fsize(pc),a0
         move.l  d4,(a0)
         move.w  #$9170,d1
         move.l  #$03_00_00_06,d2
         move.l  #32*2,d3
         lea.l   fname(pc),a1
         IOCS    _B_WRITE              *Human のファイルとなる
return:  bsr     b_eject
exit:    move.w  #$9100,d1
         move.w  dskmd(pc),d3          *プログラム実行時のドライブ状態
         btst.l  #1,d3                 *ディスクは入っていたか?
         beq     nodisk
         moveq.l #4,d2                 *入っていたなら
         IOCS    _B_DRVCHK             *挿入を促す
         moveq.l #0,d2
wait_d:  IOCS    _B_DRVCHK             *挿入を待つ
         btst.l  #1,d0
         beq     wait_d
nodisk:  btst.l  #7,d3                 *ディスク非挿入時 LED は点滅?
         bne     wink_l
         moveq.l #5,d2
         IOCS    _B_DRVCHK
wink_l:  btst.l  #6,d3
         beq     ok_ejt
         moveq.l #2,d3
         IOCS    _B_DRVCHK
ok_ejt:  ori.w   #$700,sr              *割り込み禁止
k_flsh:  IOCS    _B_KEYSNS
         tst.w   d0
         beq     nokey
         IOCS    _B_KEYINP
         bra     k_flsh
nokey:   bsr     ld_mfp
         movem.l (sp)+,d0-d7/a0-a2
         move.l  spsave(pc),sp
         clr.b   $E8E00D               *SRAM 書き込み禁止
         rte
*-----------------------------------
off:     ori.w   #$700,sr              *割り込み禁止
         bsr     ld_mfp
         movem.l (sp)+,d0-d7/a0-a2
         move.l  spsave(pc),sp
         clr.b   $E8E00D               *SRAM 書き込み禁止
         move.l  lstvct(pc),-(sp)
         rts
*-----------------------------------
sv_mfp:  movea.l #$E88009,a0           *割り込みイネーブル
         lea.l   mfp(pc),a1
         move.b  (a0),(a1)+
         andi.b  #%111_1000,(a0)       *電源を切らせない
         movea.l #$E88013,a0           *割り込みマスク
         move.b  (a0),(a1)+
         move.b  #%0001_1110,(a0)      *キーボードは生かしておく
         addq.l  #2,a0
         move.b  (a0),(a1)
         clr.b   (a0)
         rts
*-----------------------------------
ld_mfp:  movea.l #$E88013,a0           *割り込みマスクを復帰する
         lea.l   mfp(pc),a1
         move.b  1(a1),(a0)
         addq.l  #2,a0
         move.b  2(a1),(a0)
         movea.l #$E88009,a0
         move.b  (a1),(a0)             *割り込みイネーブルの復帰
         rts
*-----------------------------------
b_eject:
         move.w  #$9100,d1
eject:   IOCS    _B_EJECT              *EJECT
         moveq.l #0,d2
eject0:  IOCS    _B_DRVCHK             *EJECT を待つ
         btst.l  #1,d0
         bne     eject0
         rts
*-----------------------------------
write:   bsr     nxtsec
         tst.w   d0
         bmi     return
         IOCS    _B_WRITE
         btst.l  #30,d0
         bne     wrt_er
         clr.w   (a2)+
         rts
wrt_er:  move.w  #$FF7,(a2)+           *不良セクタ
         moveq.l #-1,d0
         rts
*-----------------------------------
nxtsec:  moveq.l #0,d0                 *次のセクタを得る
         addq.l  #1,d2
         cmpi.b  #8,d2
         bls     nsec0
         move.b  #1,d2
         bchg.l  #8,d2
         beq     nsec0
         addi.l  #$1_0000,d2
         cmpi.l  #$034C_0108,d2
         bls     nsec0
         moveq.l #-1,d0
nsec0:   rts
*-----------------------------------
         .data
         .even
spsave:  .dc.l   0
dskmd:   .dc.w   0
buff:    .dcb.b  1024,0
fat2:    .dcb.b  1024*3,0     *必要バイト数  (16*77-11)*2     =2442
fat1_5:  .dcb.b  1024*2,0     *   "          (16*77-11)*3/2+3=1835.5
fname:   .dc.b   'SCREEN  ','DAT'
attrib:  .dc.b   32
fname2:  .dcb.b  10,0
time:    .dc.w   %000_00000_01100_000              *12:00:00
date:    .dc.w   %001_00001_0010100_0              *2000-01-01
seclst:  .dc.w   0
fsize:   .dc.l   0
endfat:  .dcb.b  32,0
mfp:     .dcb.b  3,0
         .even
stacktop:
         .dcb.b  1024,0
stackbtm:
         .end
```

リスト3. LOADSC. S

```
         .include        doscall.mac
         .include        iocscall.mac
         .include        screen.equ
CR       equ     $0D
LF       equ     $0A
EOS      equ     $00
SPACE    equ     $20
TAB      equ     $09
ESC      equ     $1B
STDERR   equ     $02
STK_SIZ  equ     1024*4
BUFF     reg     a5
         .offset 0
POS:     .ds.w   2*5
T_LIM:   .ds.w   2                     *TEXT PLANE 表示開始座標の上限
XLEN:    .ds.w   1                     *表示画面 X
YLEN:    .ds.w   1                     *   "     Y
SCRMOD:  .ds.w   1                     *実画面
PAGE:    .ds.b   1                     *頁数
         .text
         .even
ent:     lea.l   16(a0),a0
         adda.l  #STK_SIZ,a1
         movea.l a1,sp
         sub.l   a0,a1
         move.l  a1,-(sp)
         move.l  a0,-(sp)
         DOS     _SETBLOCK
         addq.l  #8,sp
         move.l  #512*1024,d1
         moveq.l #1,d2
getmem:  move.l  d1,-(sp)
         DOS     _MALLOC
         addq.l  #4,sp
         tst.l   d0
         bpl     gotmem
         lsr.l   #1,d1
         cmpi.l  #1024,d1
         bmi     memerr
         lsl.l   #1,d2
         bra     getmem
gotmem:  movea.l d0,BUFF
         lea.l   bufsiz(pc),a0
         move.l  d1,(a0)
         move.w  d2,4(a0)
         movea.l BUFF,a1
         tst.b   (a2)+
         bne     getarg
         lea.l   fname(pc),a0
defflp:  move.b  (a0)+,(a1)+           *ファイル名の指定が無い時
         bne     defflp                *デフォルトのファイルネームを使用
         bra     open
getarg:
         lea.l   option(pc),a0
garglp:  tst.b   (a2)
         beq     argend
         cmpi.b  #'/',(a2)
         beq     getsw
         cmpi.b  #'-',(a2)
         beq     getsw
         cmpi.b  #SPACE,(a2)
         beq     dlmt
         cmpi.b  #TAB,(a2)
         beq     dlmt
         move.b  (a2)+,(a1)+           *command line をバッファにコピー
         bra     garglp
dlmt:    bsr     skipsp                *ファイル名指定は1つなので
         tst.b   (a2)                  *空白文字の後は、
         beq     argend                *EOF, /, - のどれか
         cmpi.b  #'/',(a2)             *でなくてはならない
         beq     getsw                 *
         cmpi.b  #'-',(a2)             *
         beq     getsw                 *
         bra     usage
getsw:   addq.l  #1,a2
         andi.b  #%1101_1111,(a2)      *UPPER
         cmpi.b  #'S',(a2)+
         bne     usage
         tst.b   (a0)
         bmi     usage                 */S を2回指定した
         move.b  #-1,(a0)
         bsr     skipsp
         bra     garglp
argend:  clr.b   (a1)+                 *EOS
         pea.l   256(BUFF)
         pea.l   (BUFF)
         DOS     _NAMECK               *command line の
         addq.l  #8,sp                 *内容をチェック
         tst.l   d0
         beq     open                  *ファイル名を指定している
         cmpi.l  #$FF,d0
         beq     path                  *パス名の指定だった
         bra     usage                 *ワイルドカードかまったくの間違い
```

▶先日，Oh!Xを買おうとレジに向かったら，なんと前の人もOh!Xを買おうとしているではないか！ 「やった，X68000ユーザーを見つけた」と思い，声をかけるかどうか考えた。しかし，考えているすきにその人はいなくなってしまった。 一ノ瀬 宣彦(21)東京都

```
path:    movea.l BUFF,a1
pathlp:  tst.b   (a1)+                  *パス名の最後を探す
         bne     pathlp
         subq.l  #1,a1
         lea.l   fname(pc),a0
append:  move.b  (a0)+,(a1)+            *指定パスに
         bne     append                 *デフォルトのファイル名を連結
open:    move.w  #2,-(sp)                       *READ / WRITE OPEN
         pea.l   (BUFF)
         DOS     _OPEN
         addq.l  #6,sp
         tst.l   d0
         bmi     opn_er
         move.w  d0,d7                  *d7 = FILE HANDLE

         move.l  #8+20,-(sp)
         pea.l   (BUFF)
         move.w  d7,-(sp)
         DOS     _READ
         lea.l   10(sp),sp

         move.w  #$10,d1
         IOCS    _CRTMOD

         suba.l  a1,a1
         IOCS    _B_SUPER
         move.l  d0,-(sp)

         movea.l BUFF,a0
         move.w  (a0)+,d5               *d5 = CRTC R20
         move.w  d5,$E80028             *CRTC R20
         btst.l  #1,d5
         bne     vctrl
         lea.l   crtc_dat(pc),a2
         movea.l #$E80002,a1            *CRTC R01
         btst.l  #4,d5
         bne     high
         adda.l  #9*2*2,a2
high:    btst.l  #0,d5
         bne     x_512
         adda.l  #9*2,a2
x_512:   move.w  #7-1,d6
crtclp:  move.w  (a2)+,(a1)+            *CRTC R01 - R07
         dbra    d6,crtclp
         move.w  (a2)+,-8*2(a1)         *CRTC R00
         move.w  (a2),(a1)              *CRTC R08
vctrl:   move.w  (a0)+,$E82400
         move.w  (a0)+,$E82500
         move.w  (a0)+,d0
         move.w  d0,$E82600
         move.b  d0,d5                  *d5 上位:CRTC R20, 下位:VCTRL R02

         movea.l #TX_SCR,a1             *Load Scroll register (TEXT, GRAPHIC)
         moveq.l #5*2-1,d6
scr_lp:  move.w  (a0),(a1)+
         move.w  (a0)+,-(sp)            *後で使うので取っておく
         dbra    d6,scr_lp

         clr.w   -(sp)                  *file pointer = text vram data
         move.l  #1024,-(sp)
         move.w  d7,-(sp)
         DOS     _SEEK
         addq.l  #8,sp

         move.l  #1024*512,-(sp)        *LOAD DATA to TEXT VRAM
         move.l  #T_VRAM,-(sp)
         move.w  d7,-(sp)
         DOS     _READ
         lea.l   10(sp),sp

         movea.l #G_VRAM,a1
         move.l  d5,d1
         lsr.w   #8,d1
         andi.w  #$3,d1
         moveq.l #0,d2

         move.w  lp_cnt(pc),d4
         move.l  bufsiz(pc),d3
         subq.l  #1,d4
gvlp:    move.l  d3,-(sp)
         pea.l   (BUFF)
         move.w  d7,-(sp)
         DOS     _READ
         lea.l   10(sp),sp
         lea.l   (BUFF),a0

         tst.b   d1
         beq     gv16
         btst.l  #1,d1
         beq     gv256
gv65k:   move.l  d3,d6
         lsr.l   #2,d6
         subq.l  #1,d6
lp65k:   move.l  (a0)+,(a1)+
         subq.l  #1,d6
         bpl     lp65k
         dbra    d4,gvlp
         bra     loadpal

gv256:   move.l  d3,d6
         subq.l  #1,d6
lp256:   move.b  (a0)+,d2
         move.w  d2,(a1)+
         subq.l  #1,d6
         bpl     lp256
         dbra    d4,gvlp
         bra     loadpal

gv16:    move.l  d3,d6
         subq.l  #1,d6
lp16:    move.b  (a0)+,d0
         move.b  d0,d2
         lsr.b   #4,d0
         andi.w  #$F,d0
         andi.w  #$F,d2
         move.w  d0,(a1)+
         move.w  d2,(a1)+
         subq.l  #1,d6
         bpl     lp16
         dbra    d4,gvlp
         bra     loadpal

loadpal:
         move.l  #1024,-(sp)
         pea.l   (BUFF)
         move.w  d7,-(sp)
         DOS     _READ
         lea.l   10(sp),sp
         lea.l   (BUFF),a0
         movea.l #T_PAL,a1

         moveq.l #32/4-1,d6
         btst.l  #6,d5
         beq     tpallp
         moveq.l #512/4-1,d6
tpallp:  move.l  (a0)+,(a1)+
         dbra    d6,tpallp

         lea.l   512(BUFF),a0
         movea.l #G_PAL,a1
         moveq.l #32/4-1,d6
         btst.l  #8,d5
         beq     gpallp
         moveq.l #512/4-1,d6
gpallp:  move.l  (a0)+,(a1)+
         dbra    d6,gpallp

         btst.l  #6,d5
         beq     chkopt
         move.l  #1024,-(sp)
         pea.l   (BUFF)
         move.w  d7,-(sp)
         DOS     _READ
         lea.l   10(sp),sp
         lea.l   (BUFF),a0
         movea.l #SP_REG,a1

         move.w  #1024/2-1,d6
spr_lp:  move.w  (a0)+,(a1)+
         dbra    d6,spr_lp

         move.l  #18,-(sp)
         pea.l   (BUFF)
         move.w  d7,-(sp)
         DOS     _READ
         lea.l   10(sp),sp
         lea.l   (BUFF),a0
         movea.l #BG_REG,a1

         moveq.l #18/2-1,d6
bg_lp:   move.w  (a0)+,(a1)+
         dbra    d6,bg_lp

         move.w  #1,-(sp)
         move.l  #1024-18,-(sp)
         move.w  d7,-(sp)
         DOS     _SEEK
         addq.l  #8,sp

         movea.l #PCG,a1
         moveq.l #32-1,d6
pcglp0:  move.l  #1024,-(sp)
         pea.l   (BUFF)
         move.w  d7,-(sp)
         DOS     _READ
         lea.l   10(sp),sp
         lea.l   (BUFF),a0

         move.w  #1024/4-1,d5
pcglp1:  move.l  (a0)+,(a1)+
         dbra    d5,pcglp1
         dbra    d6,pcglp0

chkopt:  btst.b  #0,option(pc)          */S の指定が無ければここまで
         bne     initsc
         lea.l   20(sp),sp
         bra     exit

initsc:  move.w  CRTC_20,d6             *d6 = CRTC R20
         lea.l   T_LIM(BUFF),a0
         move.w  #5*2-1,d0
scdat:   move.w  (sp)+,-(a0)
         dbra    d0,scdat

         lea.l   XLEN(BUFF),a0
setx:    btst.l  #1,d6
         bne     x768
         btst.l  #0,d6
         bne     x512
x256:    move.w  #256-1,(a0)+
         bra     sety
x512:    move.w  #512-1,(a0)+
         bra     sety
x768:    move.w  #768-1,(a0)+

sety:    btst.l  #2,d6
         bne     y512
y256:    move.w  #256-1,(a0)+
         bra     setmod
y512:    move.w  #512-1,(a0)+

setmod:  btst.l  #10,d6
         bne     md1024
md512:   move.w  #512-1,(a0)+
         bra     setpg
md1024:  move.w  #1024-1,(a0)+
         move.b  #1,(a0)+
         bra     setlim

setpg:   btst.l  #9,d6
         bne     pg1
         btst.l  #8,d6
         bne     pg2
pg4:     move.b  #4-1,(a0)+
         bra     setlim
pg2:     move.b  #2-1,(a0)+
         bra     setlim
pg1:     move.b  #1-1,(a0)+

setlim:  move.w  #1024-1,d0             *TEXT Xスクロール上限
         move.w  XLEN(BUFF),d1
         sub.w   d1,d0
         move.w  d0,T_LIM(BUFF)

         moveq.l #0,d6                  *d6.b   0=text, 1=g0, 2=g1...
         lea.l   POS(BUFF),a0
         movea.l #TX_SCR,a1


kflush:  IOCS    _B_KEYSNS
         tst.w   d0
         beq     loop
         IOCS    _B_KEYINP
         bra     kflush
loop:    IOCS    _B_KEYINP
         cmpi.b  #ESC,d0
         beq     exit
```

▶先日，新聞の「ライオン脱走」という見出しを見つけた。「ライオン脱走ったって，西武はとっくの昔に独走態勢に入ってるだろ」と思ってよく見たら本当のことだった。毎○新聞をスポーツ新聞と勘違いしていたのね。　小井田　伸雄(19)東京都

```
        cmpi.b  #SPACE,d0
        bne     upper
        bsr     nxt_pg
upper:  andi.b  #%1101_1111,d0
        cmpi.b  #'S',d0
        beq     save
curmv:
        moveq.l #7,d1
        IOCS    _BITSNS
        move.b  d0,d1
        andi.b  #%0111_1000,d1
        beq     loop
        btst.l  #3,d1
        beq     chk_up
        bsr     left
chk_up: btst.l  #4,d1
        beq     chk_r
        bsr     up
chk_r:  btst.l  #5,d1
        beq     chk_dn
        bsr     right
chk_dn: btst.l  #6,d1
        beq     loop
        bsr     down
        bra     loop

exit:   move.l  (sp)+,a1
        IOCS    _B_SUPER
        move.w  d7,-(sp)
        DOS     _CLOSE
        DOS     _EXIT
*-----------------------------------------
skpsp0: addq.l  #1,a2
skipsp: cmpi.b  #SPACE,(a2)
        beq     skpsp0
        cmpi.b  #TAB,(a2)
        beq     skpsp0
        rts
*-----------------------------------------
nxt_pg: addq.l  #1,d6
        addq.l  #4,a0
        move.b  PAGE(BUFF),d0
        addq.l  #1,d0
        cmpi.b  #2,d0
        bne     nxtpg0
        cmpi.b  #1,d6
        beq     nxtpg0
        addq.l  #4,a0
nxtpg0: cmp.b   d0,d6
        bls     okpage
        moveq.l #0,d6
        lea.l   POS(BUFF),a0
okpage: rts
*-----------------------------------------
left:   movem.w (a0),d2-d3
        addq.l  #1,d2
        bsr     chkpos
        bsr     scroll
        movem.w d2-d3,(a0)
        rts
*-----------------------------------------
up:     movem.w (a0),d2-d3
        addq.l  #1,d3
        bsr     chkpos
        bsr     scroll
        movem.w d2-d3,(a0)
        rts
*-----------------------------------------
right:  movem.w (a0),d2-d3
        subq.l  #1,d2
        bsr     chkpos
        bsr     scroll
        movem.w d2-d3,(a0)
        rts
*-----------------------------------------
down:   movem.w (a0),d2-d3
        subq.l  #1,d3
        bsr     chkpos
        bsr     scroll
        movem.w d2-d3,(a0)
        rts
*-----------------------------------------
scroll: move.w  d6,d4
        add.w   d4,d4
        add.w   d4,d4
        cmpi.b  #2,d6
        bne     setscr
        cmpi.b  #2-1,PAGE(BUFF)
        bne     setscr
        add.w   #4,d4
setscr: lea.l   0(a1,d4),a2
        move.w  d2,(a2)
        move.w  d3,2(a2)
        tst.w   d6
        beq     scrl0
        cmpi.b  #4-1,PAGE(BUFF)
        beq     scrl0
        move.w  d2,4(a2)
        move.w  d3,6(a2)
        cmpi.b  #2-1,PAGE(BUFF)
        beq     scrl0
        move.w  d2,8(a2)
        move.w  d3,$A(a2)
        move.w  d2,$C(a2)
        move.w  d3,$E(a2)
scrl0:  IOCS    _B_SFTSNS
        btst.l  #2,d0
        beq     scrl1
        move.w  #$F000,d5
wait:   nop
        dbra    d5,wait
scrl1:  rts
*-----------------------------------------
chkpos: tst.b   d6
        beq     txtpos
chkx0:  tst.w   d2
        bpl     chkx1
        move.w  SCRMOD(BUFF),d2
chkx1:  cmp.w   SCRMOD(BUFF),d2
        bls     chky0
        moveq.l #0,d2
chky0:  tst.w   d3
        bpl     chky1
        move.w  SCRMOD(BUFF),d3
chky1:  cmp.w   SCRMOD(BUFF),d3
        bls     chkps0
        moveq.l #0,d3
txtpos: tst.w   d2
        bpl     chktx0
        moveq.l #0,d2
chktx0: cmp.w   T_LIM(BUFF),d2
        bls     chkty0
        subq.l  #1,d2
chkty0: tst.w   d3
        bpl     chkty1
        move.w  #1024-1,d3
chkty1: cmpi.w  #1024-1,d3
        bls     chkps0
        moveq.l #0,d3
chkps0: rts
*-----------------------------------------
save:   clr.w   -(sp)
        move.l  #4*2,-(sp)
        move.w  d7,-(sp)
        DOS     _SEEK
        addq.l  #8,sp

        move.b  PAGE(BUFF),d0
        cmpi.b  #4-1,d0
        beq     svscr

        lea.l   POS+4(BUFF),a2          *a2=GRAPHIC #0
        move.l  (a2),4(a2)
        cmpi.b  #2-1,d0
        beq     save0
        move.l  (a2),8(a2)
save0:  move.l  8(a2),$C(a2)
svscr:  move.l  #5*4,-(sp)
        pea.l   POS(BUFF)
        move.w  d7,-(sp)
        DOS     _WRITE
        addq.l  #8,sp
        bra     loop
*-----------------------------------------
*
*       エラーメッセージの表示
*
opn_er: lea.l   openmes(pc),a0
        subq.l  #1,a1
mkmes:  move.b  (a0)+,(a1)+
        bne     mkmes
        lea.l   (BUFF),a0
        bra     putmes
memerr: lea.l   memmes(pc),a0
        bra     putmes
usage:  lea.l   usagemes(pc),a0
putmes: move.w  #STDERR,-(sp)
        pea.l   title(pc)
        DOS     _FPUTS
        addq.l  #4,sp
        pea.l   (a0)
        DOS     _FPUTS
        addq.l  #6,sp
        move.w  #1,-(sp)
        DOS     _EXIT2
*-----------------------------------------
        .data
        .even

*               R01 R02 R03 R04  R05 R06 R07  R00 R08
crtc_dat:
        .dc.w   $09,$11,$51,$237,$05,$28,$228,$5B,$1B   *高解像度       512*5
12, 512*256
        .dc.w   $04,$06,$26,$237,$05,$28,$228,$2D,$1B   *               256
*256
        .dc.w   $03,$05,$45,$103,$02,$10,$100,$4B,$2C   *低解像度       512*5
12, 512*256
        .dc.w   $01,$00,$20,$103,$02,$10,$100,$25,$24   *               256
*256
fname:  .dc.b   'SCREEN.DAT',0
title:
        .dc.b   'LOADSC.X ver1.00 1992 1mRicky Sugimoto1m',CR,LF,0
openmes:
        .dc.b   ' が見付かりません',CR,LF,0
memmes: .dc.b   'メモリが足りません',CR,LF,0
usagemes:
        .dc.b   '使用法:LOADSC [/S] [FILENAME]',CR,LF
        .dc.b   TAB,'SAVESC 形式のファイルを表示します',CR,LF
        .dc.b   TAB,'ファイル名省略時は、SCREEN.DAT をカレントディレクトリから探します',CR,LF
        .dc.b   TAB,'/S を指定すると表示後 Graphic / Text 画面の表示位置を変更可能です',CR
,LF
        .dc.b   TAB,'S キーを押すとこの時点の表示位置をファイルに書き込みます',CR,LF,0
option: .dc.b   0

        .even                   *sp を偶数アドレスに設定するため
bufsiz: .dc.l   0
lp_cnt: .dc.w   0

        .end
```

リスト4

```
0000   20 D0 2D 6C 68 31 2D 97   : E6
0008   05 00 00 E4 22 00 00 5D   : 68
0010   AF E7 18 20 00 0A 73 61   : AC
0018   76 65 73 63 2E 73 79 73   : 3E
0020   84 04 EA 78 71 98 78 1E   : 89
0028   73 1C C8 E0 1E 1C 8F F2   : F2
0030   EB B5 39 C8 C0 8C 68 03   : 58
0038   29 70 03 D4 48 02 E8 80   : 22
0040   1E CB 2C C7 62 B9 A6 4F   : EC
0048   EB 02 26 E8 D9 99 B9 48   : 6E

0050   9E F7 9D C5 9E 06 75 12   : 22
0058   8E 5F 3E 1A F3 D3 77 AD   : 2F
0060   ED 76 F7 DC 88 E4 E1 64   : E7
0068   26 66 2B 63 10 BF 9F 13   : 9B
0070   3D 18 01 73 9B 70 BB CC   : 5B
0078   6C 64 E4 36 10 81 A7 F6   : 18
-------------------------------------
SUM:   46 DC DA 3D 5E AF 9D EA   15E8

0080   82 5F 6C ED 6D 83 16 C5   : 05
0088   28 31 9A F6 02 B6 AC 08   : 55
0090   BD 73 D3 19 F5 D9 A7 DF   : 70
0098   B2 BC B3 41 15 8E 84 ED   : 76
00A0   F5 5F EB EE EC 27 55 FE   : 93
00A8   64 BD 24 FA C5 F2 B7 1F   : CC
00B0   00 D0 1D 3A 9C 92 04 30   : 89
00B8   EE 2B B1 71 C7 AA 3C 88   : 70
00C0   94 D1 A7 E8 CE 49 3D 17   : 5F
00C8   E1 97 9F FD A6 CF 72 F8   : F3
00D0   E4 53 8C A2 15 2E 36 62   : 40
```

▶10月号での「MUSIC PRO-68K[MIDI]と仲よくなろう」を読んで，なんだか懐かしくなりました。以前，僕も使っていて必死に打ち込んでいたっけ。いまでは「Mu-1 Super」という強い味方がいるのでとても楽になりました。でも早く「MUSIC PRO-68K[MIDI]」がバージョンアップしないかな。

久保木　稔(20)東京都

```
00D8   F4 A3 14 E3 4E F2 80 90   : DE
00E0   AC AA 44 DE EF 8C 95 95   : 1D
00E8   E9 4C 01 D1 04 90 F6 83   : 14
00F0   5C 62 FD 95 55 8C 5F FF   : 8F
00F8   B9 FA DF A9 09 22 75 F3   : CE
---------------------------------------
SUM:   57 86 70 27 B5 F7 FD 79   CCE1

0100   1F 7C 23 9F 3B 3B 34 DE   : E5
0108   FC AC 61 F9 4D D4 29 D0   : 1C
0110   6A 5D 5E 51 04 97 F3 BB   : BF
0118   EF 12 F2 D6 BE CF 3E 28   : BC
0120   82 EF 6B F6 FC 2F 41 13   : 51
0128   E6 60 2F 3F CE 40 5F 46   : 67
0130   64 BB 7D A0 B7 F5 C5 C8   : 75
0138   01 69 9B 61 6E C1 9E C1   : F4
0140   EB 77 01 02 FF 1F 6B ED   : DB
0148   7F 64 C3 5C 0B 93 5E CD   : CB
0150   C3 E3 5D F1 7F EF 56 2E   : E6
0158   5E DD F1 93 0B 0F 9C 02   : 77
0160   D7 95 BC 3F DC 9F 37 EC   : 05
0168   E6 F2 A7 5E 35 62 FC 5F   : CF
0170   3F 46 32 6A B1 1A 80 17   : 83
0178   5C 70 14 C6 A9 22 E8 07   : 60
---------------------------------------
SUM:   24 E2 41 A4 38 87 E7 C6   9668

0180   A3 C9 27 D9 CE 65 F1 6F   : FF
0188   E1 45 4D D3 7B 04 F0 54   : 09
0190   B4 DE 0A 2C 35 28 4B 7F   : EF
0198   A0 62 3C 89 6C 05 EF C2   : E9
01A0   D3 E2 D6 FB 22 AF EE 39   : 7E
01A8   02 86 62 A0 51 85 51 90   : 41
01B0   5F BC C8 E1 69 B1 6E 05   : 51
01B8   A2 9A 8B 16 69 F7 3E 65   : E0
01C0   E9 5D F0 AB E1 69 C6 77   : 68
01C8   9E 77 96 6A 49 B0 D1 92   : 71
01D0   30 19 B7 C0 67 D3 6D DF   : 46
01D8   B0 B9 D4 45 35 B6 14 4B   : CC
01E0   AE 23 28 79 22 26 D7 03   : 94
01E8   11 05 21 8E 49 7F 00 C6   : 53
01F0   7C EB B0 84 F6 D7 FE A3   : 09
01F8   FC 01 D7 F7 85 E0 BE B9   : A7
---------------------------------------
SUM:   4C C6 26 8F DB 70 B1 8F   9D6F

0200   59 35 A7 B3 93 5A A6 59   : D4
0208   2C 06 B9 7D 80 C8 B3 0C   : 6F
0210   1E DC C4 3D B8 48 77 F9   : 6B
0218   57 5E 19 1D FD 10 F4 C6   : B2
0220   90 77 E6 6C 9A D3 60 E9   : 0F
0228   DE 7B 1E 3B E8 D8 C7 71   : AA
0230   83 E3 B2 5B EB 74 32 09   : 0D
0238   69 CA A1 2C BF A9 75 8D   : 6A
0240   A3 52 53 04 AE 93 E5 A9   : 1B
0248   41 FD 1B 8D AB 8D 6C 70   : FA
0250   A5 F4 4B 5D 74 95 0A F3   : 47
0258   08 A9 69 06 27 5F 95 42   : 7D
0260   76 00 81 3A 0F DA 53 93   : 00
0268   F7 22 13 E0 2A 22 43 81   : 1C
0270   A9 A0 FC B3 BD 68 66 31   : B4
0278   26 86 60 13 8A F5 8D D0   : FB
---------------------------------------
SUM:   21 48 A6 8C 68 AF 0B 77   5AEB

0280   1F C4 80 3D 09 53 CB 28   : EF
0288   12 BA BD E2 9D 9C 40 32   : 16
0290   9D C5 C5 53 42 0A B0 CB   : 41
0298   FD 67 14 B8 55 13 D0 35   : 9D
02A0   BA AF 38 47 9B 15 65 67   : 64
02A8   50 30 59 23 5D 3B 8F 1B   : 3E
02B0   F3 F5 CC 6A 32 AC 32 93   : C1
02B8   22 D5 BC 8F 7C 47 FF 7A   : 7E
02C0   14 42 FB A8 18 13 33 2D   : 84
02C8   A9 AD 92 9D 9B D5 63 12   : 6A
02D0   95 95 0D 26 AF 50 BC B5   : CD
02D8   7C 28 A5 10 10 CD 4F 67   : EC
02E0   3F 89 04 EB 9C 63 9B E4   : 35
02E8   C2 AA 4C 2A 17 AD F7 33   : D0
02F0   F3 09 C6 8C D4 53 E2 CA   : 21
02F8   16 50 E7 9F C7 A2 08 13   : 70
---------------------------------------
SUM:   C2 8B 6B 48 A3 59 CD 38   4DEB

0300   18 A9 8A 78 C1 4A DB CB   : 74
0308   37 7D 85 59 6F B3 10 22   : E6
0310   C7 C6 1F C7 24 AF 09 75   : C4
0318   C9 30 64 FE DA 9E 71 FC   : 40
0320   33 AE 5F F5 F8 91 9D FF   : 5A
0328   71 8A 70 64 A0 D2 54 FF   : 94
0330   56 60 75 E1 BE 68 C8 41   : 3B
0338   8C E7 DD 6D C2 71 7F 68   : D7
0340   2F C7 F6 EF 63 0A 7D D4   : 99
0348   4B 71 92 2F 6B 57 F1 85   : B5
0350   EA 38 04 38 CF 32 7B A5   : 7F
0358   2A AF 5E AB CC 61 7A 3B   : C4
0360   F7 F3 C6 73 05 F1 1F B7   : EF
0368   88 F8 05 AF DE 55 B6 28   : 45
0370   5F BF 81 CF 1B 92 83 2F   : CD
```

```
0378   4F B9 D4 FA 4E 2F 0B 8F   : ED
---------------------------------------
SUM:   20 1D BD 29 FB 81 63 DB   F8C5

0380   02 30 FB 9D C0 A9 22 82   : D7
0388   B3 2D 65 97 F9 BC 61 B1   : A3
0390   F6 96 50 C7 2C 0C DD 14   : CC
0398   96 8F EF 83 F9 C3 8D A6   : 86
03A0   82 D4 A1 BD 09 E4 DE 83   : 02
03A8   4B 84 E4 46 23 12 75 75   : 18
03B0   DF 40 5C AF 1F E9 1F 12   : 63
03B8   DA F4 8B 53 9E 2D E4 75   : D0
03C0   EF 38 7E CD 34 C3 4D 46   : FC
03C8   63 E6 61 1D A0 9D FE F1   : F3
03D0   6D 26 5F F7 1B 6A 01 76   : E5
03D8   4C 9C 4F C9 21 51 54 B6   : 7C
03E0   2A 2F 2D 3B 97 6A 31 D3   : C6
03E8   3D CA 7B BA FD 06 E3 C4   : E6
03F0   4E AE A1 A6 FF 9B 03 4F   : 2F
03F8   E0 4B D3 6F 06 79 47 C9   : FC
---------------------------------------
SUM:   67 E0 B4 37 70 DF 41 7E   E5A3

0400   23 49 9B F4 01 30 21 FB   : 48
0408   B3 FC AB 92 84 9B AA 56   : 0B
0410   F2 79 D1 80 57 81 B5 58   : A1
0418   74 FB 44 6D DE 84 3F EF   : B0
0420   EC 22 B6 D1 E5 B7 12 9A   : DD
0428   74 2B 69 57 E4 13 20 7E   : F4
0430   45 05 39 64 B8 E2 77 5D   : 55
0438   02 5F B1 98 FF EF 2B C0   : 83
0440   57 19 5E EF E4 B8 BE FE   : 15
0448   BC 34 9D C9 F3 9F EA 9F   : 71
0450   47 56 4D A9 58 06 44 40   : 75
0458   43 FB 78 87 D0 80 DC 61   : CA
0460   ED A7 F1 F4 FB 32 C7 E3   : 50
0468   05 80 C9 FF 2D CA 35 92   : 0B
0470   FB C2 04 05 BC 88 0B 02   : 17
0478   B9 B8 F8 60 A1 27 E2 B8   : 2B
---------------------------------------
SUM:   26 A9 DA D7 BE F3 44 3A   0D97

0480   F4 98 37 48 18 EF 35 68   : AF
0488   35 E1 7D 4F 91 8A 3F D8   : 14
0490   FC 62 BD 5F 0B FB AD F1   : 1E
0498   93 89 56 3D 8D 3F 27 77   : 19
04A0   CB 6B 09 C2 B6 CE 79 40   : 3E
04A8   FC 0F 52 53 B9 E4 9D FF   : E9
04B0   CB 48 B0 28 01 CA 00 12   : C8
04B8   60 0D 68 01 C8 01 2E 00   : CD
04C0   CB 00 74 00 0C 80 1D 00   : E8
04C8   25 00 2A 80 32 00 35 00   : 36
04D0   38 80 39 80 0B 00 22 00   : 9E
04D8   4C 00 E0 02 40 04 C0 0A   : 3C
04E0   40 15 00 31 00 62 00 D0   : B8
04E8   01 A0 03 70 07 00 0E 00   : 29
04F0   1C 80 08 00 28 00 E0 04   : B0
04F8   00 10 00 58 01 60 06 80   : 4F
---------------------------------------
SUM:   7B F8 FC 6C 32 76 B4 57   F1B5

0500   22 00 88 02 20 09 00 26   : FB
0508   00 A0 02 80 0A 00 2E 00   : 5A
0510   B8 03 00 0C 00 32 00 C8   : C1
0518   03 20 0C 80 34 00 D8 03   : BE
0520   60 0D 80 36 00 D8 03 60   : 5E
0528   0E 00 04 00 20 01 00 10   : 43
0530   00 80 06 00 30 01 80 0C   : 43
0538   00 60 04 00 20 01 40 0C   : D1
0540   00 60 03 00 1C 00 E0 07   : 66
0548   00 40 02 00 10 00 80 04   : D6
0550   00 20 01 20 09 00 48 02   : 94
0558   40 12 00 90 04 80 28 01   : 8F
0560   40 0B 00 58 02 C0 16 00   : 7B
0568   E0 E0 06 B2 34 AB 11 43   : AB
0570   BB 88 E5 41 11 53 F1 85   : 43
0578   A0 50 57 AC 95 B0 05 80   : BD
---------------------------------------
SUM:   06 45 6C EB E3 04 B6 CF   97F8

0580   2C 01 60 0B 00 58 02 C0   : B2
0588   16 00 B0 06 00 30 01 80   : 7D
0590   0C 00 60 03 00 18 00 F6   : 7D
0598   00 09 41 D1 9F 4A 6E CA   : 3C
05A0   D6 9F 3F 28 7D B4 BB 04   : CC
05A8   B0 55 9A 44 2D 05 44 F1   : 4A
05B0   7F 27 0A D3 7C 0F AD 24   : DF
05B8   AC 1E 33 2D 6C 68 31 2D   : 5C
05C0   48 05 00 00 66 07 00 00   : BA
05C8   14 B5 DD 18 20 00 08 6C   : 52
05D0   6F 61 64 73 63 2E 78 C4   : 74
05D8   54 EA 78 71 92 80 0C 8F   : D4
05E0   E3 0E FE 35 1A 54 23 52   : 07
05E8   00 E6 BA 5F CE 2F C4 59   : 19
05F0   65 BD 9B AE A8 FA A6 EF   : A2
05F8   2B 59 94 8B EB 77 06 F5   : 00
---------------------------------------
SUM:   91 52 67 1A 27 C3 6D 94   1C93
```

```
0600   DC 8B 6A A7 30 0B DA 32   : BF
0608   6A 71 7C 94 D5 01 9F 69   : C9
0610   AD C2 B9 80 D8 C7 21 1F   : 87
0618   BC B0 74 8D E2 DE C7 2D   : 21
0620   B7 32 E7 15 92 B4 6B 25   : BB
0628   B7 BC E8 4A D2 F6 5B E9   : B1
0630   BE 52 EE E3 1C FD 33 D6   : 03
0638   C7 AB 19 08 20 A0 A0 6D   : 60
0640   4D F3 A7 CF BC CD 5A 51   : EA
0648   6F B0 71 02 EE 54 ED 13   : D4
0650   81 62 5A 65 99 01 6A CB   : 71
0658   4C D6 40 B7 05 ED 2C 7A   : B1
0660   F3 84 77 A0 E9 D9 E0 BB   : EB
0668   3A 2C 46 21 30 CD D2 38   : D4
0670   E0 C5 EA D9 EC 7B E6 C4   : 79
0678   B7 D0 33 65 05 36 A0 0D   : 07
---------------------------------------
SUM:   EF 79 75 7E B1 5E 0F A5   343C

0680   C0 C8 C7 17 EF 04 BF 3F   : 57
0688   E9 47 42 68 12 3C 4F 9C   : 13
0690   3A 74 0D 73 B5 1E 67 50   : B8
0698   41 3E AC 8E 52 DD 98 E8   : 68
06A0   AA 8D 27 CC 3C 0C A3 D1   : E6
06A8   68 C9 B5 D2 12 BF 42 9B   : 66
06B0   4B C3 A1 4F 44 28 CB 51   : 86
06B8   7D ED 08 B4 EB 6F E1 9E   : FF
06C0   4C AE 37 DC DC 58 54 C5   : 5A
06C8   87 A2 F3 FB E8 8A BA 9B   : DE
06D0   FB F3 0A DE FC 1B CF D4   : 90
06D8   29 35 3D CC 2E 01 9B 8D   : BE
06E0   DD EA D2 40 59 37 8B E3   : D7
06E8   B4 35 F0 3F B3 E0 7E 3A   : 63
06F0   62 CB 3F 2A 9B 62 54 17   : FE
06F8   71 60 5F 4C 18 EF 37 41   : FB
---------------------------------------
SUM:   59 89 18 97 32 03 AA A4   9493

0700   82 6B 90 28 E7 F5 B5 52   : 88
0708   0F CA 51 86 E7 5A DE 9A   : 69
0710   CD 02 B1 69 27 63 BD 62   : 92
0718   BB 98 D5 BD 79 E4 19 C3   : 1E
0720   D7 B9 4A 2C 4B AE 42 B9   : FA
0728   99 07 F6 99 37 54 8C 7E   : C4
0730   6D 7C F9 02 E6 BB 6C 05   : F6
0738   5D A8 A1 1E 93 66 C3 1C   : 9C
0740   1A AF 79 10 FE FF 68 B4   : 6B
0748   B6 A7 45 0F F4 D1 B7 46   : 73
0750   4F 58 22 B9 84 2C AF F8   : D9
0758   7F CB F5 55 96 EF DB F7   : EB
0760   35 EC 77 5F E9 B7 B2 51   : 9A
0768   6E 1A C1 C4 6C C7 B8 9F   : 97
0770   E3 05 BB 0F 9B 64 D5 B7   : 3D
0778   89 A7 D0 77 5F FC A0 47   : B9
---------------------------------------
SUM:   00 DE D9 8F C4 82 EE 40   F82B

0780   D6 A8 47 DB A2 2C E0 88   : D6
0788   EF E0 29 6A D6 85 35 E5   : D7
0790   BB 63 4E 9C 1C 6C 03 E5   : 78
0798   8B 07 FE CB 33 99 FA 31   : 52
07A0   AF DB 28 6F 1F 77 E9 FC   : 9C
07A8   8E C9 AF 9B 59 30 BD DC   : C3
07B0   5A 98 D2 DC 18 52 2E 11   : 49
07B8   51 18 60 CB CD A0 66 60   : C7
07C0   9F 77 D1 90 FB 85 55 CB   : 17
07C8   A6 31 9A F8 9C 96 17 AF   : 61
07D0   9A 10 CB A5 77 19 83 F6   : 23
07D8   67 A8 9A 23 82 BC 9F 54   : FD
07E0   A7 87 64 69 76 56 00 26   : ED
07E8   78 0B A6 2C 5D F9 D1 1E   : 9A
07F0   B0 DF F6 39 9B 07 D6 E5   : 1B
07F8   16 01 BC B0 44 67 86 E2   : 96
---------------------------------------
SUM:   1E 18 51 2B 66 FC 07 9B   8F4F

0800   29 B4 FC 17 A9 2A CC 8C   : 1B
0808   F3 F9 A7 30 AF 8A 1F B5   : D0
0810   0D 03 00 D8 48 00 28 B4   : 0C
0818   B9 75 95 1F E5 FF B7 E7   : 64
0820   C6 1B 08 28 1B 18 85 4D   : 16
0828   B4 05 44 1E 7B E0 FA 27   : 97
0830   C1 E2 1F 21 2C 2F 41 F9   : 78
0838   2E 9A CC 41 51 63 90 28   : 41
0840   D5 42 41 18 66 84 1C 2E   : A4
0848   E1 01 66 02 7E 3D F4 BF   : B8
0850   D4 B1 6F 11 A9 E9 37 9E   : 6C
0858   22 C4 C9 C2 AF 3C B7 16   : 29
0860   FA 10 D7 50 5C BF F0 D4   : 10
0868   B4 F8 14 F7 75 AE B4 DC   : 6A
0870   40 51 5B 4E 8B 4D 95 E2   : 89
0878   8F BD CD 98 E7 64 73 A2   : 11
---------------------------------------
SUM:   74 8F 61 00 17 41 C4 46   68DD

0880   4F 66 3D 46 05 CB BB D4   : 97
0888   23 72 12 36 7F ED B2 25   : 20
0890   A9 40 3B 28 07 75 10 0F   : E7
0898   6D 00 ED E2 01 DE E7 2A   : 2C
```


▶この間，ゴミの山からカセットデッキを拾ってきた。少し直せば，と思って調べてみたがどこも悪くない。しかも，ヘッドがまったくといっていいほど汚れていないのだ。オートリバースもないような10年以上も前のものなのに，現在，家にあるカセットデッキの中でいちばんいい音がします。

佐藤　哲(21)千葉県

```
08A0  B1 C9 09 DD B5 D2 6F C4  : 1A
08A8  74 53 36 2D B6 A8 58 79  : 59
08B0  71 9E 1B 2E 2F 10 D4 FF  : 6A
08B8  2F 28 8D 7B E4 36 C9 40  : 82
08C0  2B 50 9D 6B C9 AA 91 3F  : C6
08C8  CE E5 EE 3B DB 68 9C 5C  : 17
08D0  A6 F9 E5 BD 82 17 AB 8E  : 13
08D8  4F 19 BB D3 C7 D2 EB E9  : 63
08E0  01 70 61 19 E5 02 2B C1  : BE
08E8  80 A0 A0 8B 95 52 90 AB  : 6D
08F0  4D 92 DC 1E A8 C9 F1 DD  : 18
08F8  9F F4 1F F2 EB 84 E1 F2  : E6
-----------------------------------------
SUM:  A8 D7 85 23 04 67 18 FB  7C4C

0900  AB A1 3B 7F 61 B8 31 B6  : 06
0908  BD 5B 78 BE AC 8D D3 4A  : A4
0910  56 EC 7B E0 DA CA BB 84  : 80
0918  08 49 28 2F BF 79 6F 72  : C1
0920  E1 07 24 A3 D4 09 FA 26  : AC
0928  27 BB 9A 03 CB 90 79 09  : 5C
0930  03 E2 90 3B D9 EB 92 C2  : C8
0938  73 23 59 E4 6B E8 23 46  : 8F
0940  D3 41 E8 6C 76 BF 15 43  : F5
0948  C8 65 9F 23 D6 7A A3 B5  : 97
0950  1D 6C 81 6D 82 AF 8F 96  : CD
0958  E2 38 D8 C4 DD 48 3D 85  : 9D
0960  40 3C BB 07 64 3F 21 59  : 5B
0968  0E F8 67 38 AC 74 38 76  : 73
0970  58 8E 1C AF 4B 86 FE A7  : 27
0978  CA 52 6E 60 74 98 0F 6E  : 73
-----------------------------------------
SUM:  4E 56 89 1F 03 F5 40 24  3D00

0980  95 2F 2A 57 F6 31 FB 3D  : A4
0988  8D 89 E8 AF 63 90 6C 58  : 64
0990  D2 09 26 ED 97 33 3B 90  : 83
```

```
0998  71 68 20 19 D1 E6 87 AD  : FD
09A0  9D 98 5D 37 8E 14 80 68  : 53
09A8  E1 5B 3B 6D 34 DE C9 8B  : 4A
09B0  02 41 B9 47 28 90 2A 79  : 9E
09B8  DA 77 3F E5 86 C3 1D 18  : F3
09C0  B1 31 96 38 E5 6F AC 89  : 39
09C8  77 68 FB 23 94 FE 4A F1  : CA
09D0  7B CE 7B 23 E6 BF 92 DE  : FC
09D8  64 4C 48 07 94 52 90 89  : FE
09E0  59 08 FF 90 59 03 FF FB  : 46
09E8  CE D2 FF 71 EF DA 79 53  : A5
09F0  32 E6 40 81 11 AE CE 4E  : B4
09F8  22 6A F3 27 3E DD 45 81  : 87
-----------------------------------------
SUM;  41 B1 6D 0A BB 05 5C 54  89DA

0A00  0E E0 93 3A AC 2D 3B 7E  : 4D
0A08  FD D8 90 84 70 F9 07 25  : 7E
0A10  8B 9C 7B 8A 8A F8 B4 AD  : 0F
0A18  A0 56 C0 F7 5F AD E6 D8  : 77
0A20  1E D6 D0 5B 97 CA D4 40  : 94
0A28  94 B3 F4 84 49 37 DD 99  : B5
0A30  9C 5C 7D 4B 8C 46 6A 0F  : 0B
0A38  A7 42 C7 CD 0A 52 1E 31  : 28
0A40  39 C0 4E DA 77 F0 7A 96  : 98
0A48  3F 36 7E 4B 09 F5 50 A7  : 33
0A50  E9 91 03 9A 34 6B 30 4D  : 33
0A58  1C 13 FA 6A 89 B5 1B 7E  : 6A
0A60  37 DB F2 B5 FC CB 3A AC  : 66
0A68  3E 12 CF D5 23 3B B7 53  : 5C
0A70  A9 0C 06 27 CA 4C 04 DC  : D8
0A78  27 B9 99 EC 2F 4E 8A F4  : 60
-----------------------------------------
SUM:  ED 1D 8F FC D0 09 A9 18  5E32

0A80  02 BE 4A AF D0 15 D2 17  : 87
0A88  39 0E F0 43 10 30 23 0A  : E7
```

```
0A90  02 DE A3 7F 1C CF D5 F7  : B9
0A98  A7 7C BC FA 59 7E DC E3  : 6F
0AA0  EF C9 9C A5 35 9A F4 DF  : 9B
0AA8  4C 6C 5F 76 DE 5C 68 04  : 33
0AB0  DA 80 4A F4 1B 7D 2F CB  : 2A
0AB8  EE 1F 24 1A DE F0 27 7E  : CF
0AC0  EE 2F 1D 33 79 13 C6 B4  : 73
0AC8  04 CA 49 F4 6F 98 A1 A7  : 5A
0AD0  F7 37 D8 09 36 7C 2D B9  : A7
0AD8  B1 08 70 D2 C8 F0 DF 66  : F8
0AE0  3C 05 B1 22 A4 37 E8 D8  : AF
0AE8  B7 2A 0E 62 F7 5F 3B FC  : DE
0AF0  71 D1 86 A9 ED 48 0B 20  : D1
0AF8  C5 5B 84 AE A1 BB 48 ED  : E3
-----------------------------------------
SUM:  BB 8D 79 71 70 A5 41 82  1DA8

0B00  FF 4D C8 4D C1 AE E7 36  : ED
0B08  C2 D2 2D C4 E1 A4 03 0D  : 1A
0B10  EA 01 B1 71 FB B3 FB 9A  : 50
0B18  F6 A2 9F 9A B4 BE E4 32  : 59
0B20  4A 00 00 00 00 00 00 00  : 4A
0B28  00 00 00 00 00 00 00 00  : 00
0B30  00 00 00 00 00 00 00 00  : 00
0B38  00 00 00 00 00 00 00 00  : 00
0B40  00 00 00 00 00 00 00 00  : 00
0B48  00 00 00 00 00 00 00 00  : 00
0B50  00 00 00 00 00 00 00 00  : 00
0B58  00 00 00 00 00 00 00 00  : 00
0B60  00 00 00 00 00 00 00 00  : 00
0B68  00 00 00 00 00 00 00 00  : 00
0B70  00 00 00 00 00 00 00 00  : 00
0B78  00 00 00 00 00 00 00 00  : 00
-----------------------------------------
SUM:  EB C2 45 1C 51 C3 C9 0F  D87A
```

リスト5

```
echo off
rem     ␛で始まるのは、エスケープシーケンスです。(Human68k User's Manual参照)
rem     ED.X では、CTRL+V ( CTRL を押しながら V を押す)に続いて ESC を押すと
rem     ␛を入力することができます。
rem
echo ␛[s
if "%1" == "" goto usage
if not "%3" == "" goto usage
:chkarg
if %1 == +f goto fnc+
if %1 == +F goto fnc+
if %1 == -f goto fnc-
if %1 == -F goto fnc-
if %1 == +c goto cur+
if %1 == +C goto cur+
if %1 == -c goto cur-
if %1 == -C goto cur-

:usage
echo 使用法      CON2 [SWITCH]
echo             SWITCH: +C      カーソル表示
echo                     -C      カーソル消去
echo                     +F      ファンクションキー表示
echo                     -F      ファンクションキー消去
goto exit

rem     エスケープシーケンスを " から始めているのは、
rem     > をリダイレクトでは無いと明示するため。

:cur+
echo "␛[1K␛[>5l
goto next

:cur-
echo "␛[1K␛[>5h
goto next

rem     ファンクションキー非表示モード時から表示モードへ切り替える時
rem     画面最下行にいると表示がおかしくなるので CLS によって
rem     強制的にホームポジションに戻す。
:fnc+
echo "␛[>1l
cls
shift
if "%1" == "" goto exit
goto chkarg

:fnc-
echo "␛[1K␛[>1h
:next
shift
if "%1" == "" goto end
goto chkarg

:end
echo ␛[u
:exit
```

リスト6

```
        .include        doscall.mac
        .include        iocscall.mac
        .include        const.h

TEXT0           equ     $E00000
TEXT1           equ     $E20000
TEXT2           equ     $E40000
TEXT3           equ     $E60000

        .text
        .even

ent:
        move.l  $8(a0),d0
        andi.l  #$FFFF_FFFE,d0
        movea.l d0,a7
        sub.l   a1,d0
        cmpi.l  #1024,d0
        bmi     stackerr

        move.w  #-1,-(sp)
        move.w  #14,-(sp)
        DOS     _CONCTRL
        addq.l  #4,sp
        move.l  d0,d6

        clr.l   -(sp)
        move.w  #3,-(sp)
        DOS     _CONCTRL
        addq.l  #6,sp

        clr.l   -(sp)
        DOS     _SUPER
        move.l  d0,(sp)

        movea.l #TEXT0,a0
        movea.l #TEXT1,a1
        movea.l #TEXT2,a2
        movea.l #TEXT3,a3
        clr.l   d0
        move.w  #32*4-1,d7
loop:           move.l  d0,(a0)+
        move.l  d0,(a1)+
        move.l  d0,(a2)+
        move.l  d0,(a3)+
        dbra    d7,loop

        DOS     _SUPER
        addq.l  #4,sp

        move.w  #8-1,d7
        move.w  #$0001,d1
        move.w  #1,d2
        move.w  #$000F,d3
raslp:          IOCS    _TXRASCPY
        add.l   d1,d1
        add.l   d2,d2
        dbra    d7,raslp

        move.w  d6,-(sp)
        move.w  #14,-(sp)
        DOS     _CONCTRL
        addq.l  #6,sp

        DOS     _EXIT
*-------------------------------
stackerr:
        move.w  #STDERR,-(sp)
        pea.l   stackmes(pc)
        DOS     _FPUTS
        addq.l  #6,sp
*-------------------------------
        .data
        .even
stackmes:
        .dc.b   'スタック領域が確保出来ませんでした。',CR,LF,0

        .end
```

▶10月号は全体的に面白く，笑ってばかりいた。読者も編集部もおちゃめさんが増えたのだろうか。でも師匠，ヌードはだめですよヌードは。　安藤　道子(20)宮崎県

バックナンバー案内

BACK ISSUES

ここには1991年11月号から1992年10月号までをご紹介しました。現在1991年1，5，8，9，11，12，1992年1，2，4～10月号の在庫がございます。バックナンバーおよび定期購読の申し込み方法については，168ページを参照してください。

1991

11月号
特集　空間彷徨型ゲーム大分析

連載　響子 in CGわ～るど/大人のためのX68000/ANOTHER CG WORLD
DōGA/ショートプロ/Computer Music入門/吾輩はX68000である
ようこそC言語/マシン語プログラミング/Z80's Bar/ハード工作

●X68000用カードゲーム　ギャップ
●新製品紹介　F-Card GT
LIVE in '91　オーダイン
THE SOFTOUCH　キャメルトライ/アクアレス/フューチャーウォーズ他
全機種共通システム　Small-C活用講座(応用編)/MORTAL

12月号
特集　音・そして音楽とコンピュータ
別冊付録　X68000 THE GAME SOFTWARE BEST SELECTION

連載　響子 in CGわ～るど/マシン語プログラミング/ショートプロ
ハード工作/Z80's Bar/ようこそC言語/ANOTHER CG WORLD
吾輩はX68000である/Computer Music入門/大人のためのX68000

●エレクトロニクスショウ＆データショウ
LIVE in '91　OH YEAH!/サイレント・イヴ/ジングルベル
THE SOFTOUCH　フェアリーランドストーリー/プロサッカー68他
全機種共通システム　Small-C用　SLANGコンパチ関数

1992

1月号
特集　SX-WINDOWの未来

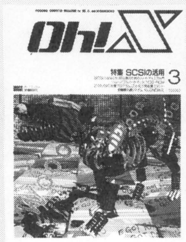
連載　響子 in CGわ～るど/DōGA・CGA/大人のためのX68000
ハード工作/Z80's Bar/ショートプロ/吾輩はX68000である
ANOTHER CG WORLD/Computer Music入門/カードゲーム

●MAGIC用ゲーム　3D MAZE
●CM-300/500&LA音源の活用法
LIVE in '92　DRAGON SABER/すき/THE ENTERTAINER
THE SOFTOUCH　出たな!! ツインビー/ブリッツクリーク/飛翔鮫他
全機種共通システム　パズルゲームLINER

2月号
特集　2Dグラフィックの拡張

連載　響子 in CGわ～るど/大人のためのX68000/マシン語プログラミング
ハード工作/ショートプロ/ANOTHER CG WORLD/Z80's Bar
吾輩はX68000である/Computer Music入門/カードゲーム

●TREND ANALYSIS
●MIRAGE MODEL STUFF/Press Conductor PRO-68K
LIVE in '92　ストリートファイターII/Tide Over
THE SOFTOUCH　ジェノサイド2/アルシャーク/コード・ゼロ他
全機種共通システム　シミュレーションゲームPOLANYI



3月号（品切れ）
特集　SCSIの活用

連載　響子 in CGわ～るど/DōGA・CGA/大人のためのX68000/Z80's Bar
ショートプロ/吾輩はX68000である/マシン語プログラミング
ハード工作/ANOTHER CG WORLD/Computer Music入門/カードゲーム

●Z-MUSIC支援ツール　ZPDCON.X
●Z's-EX用拡張コマンド　MASK_reverse.X
LIVE in '92　ギャラクシーフォース/君が代
THE SOFTOUCH グラディウスII/レミングス/大戦略III'90/伊忍道
全機種共通システム　カードゲームKLONDIKE

4月号
特集　成熟するゲームと日本の文化

連載　よい子のSX-WINDOW/Z80's Bar
響子 in CGわ～るど/ショートプロ/吾輩はX68000である
ハード工作/ANOTHER CG WORLD/Computer Music入門

●発表　1991年度GAME OF THE YEAR
●バーコードバトラー
LIVE in '92　あじさいのうた/ショパン練習曲作品25-2へ短調/IT'S MAGIC
THE SOFTOUCH　ファーストクィーンII/マスターオブモンスターズII他
全機種共通システム　実践Small-C(1)オプティマイザO80

5月号
特集　明日のための環境づくり
第7回　言わせてくれなくちゃだワ

連載　響子 in CGわ～るど/大人のためのX68000/Z80's Bar
ハード工作/ショートプロ/マシン語プログラミング
Computer Music入門/吾輩はX68000である

●製品紹介　MIDI音源 03R/W/MIC68K
LIVE in '92　フレンズ/Danger Line
THE SOFTOUCH　エイリアンシンドローム/苦胃頭捕物帳他
全機種共通システム　実践Small-C(2)COMMAND.OBJ

6月号
特別企画　Oh!MZ,Oh!X10年間の歩み
特別付録　創刊10周年記念PRO-68K(5"2HD)

連載　響子 in CGわ～るど/大人のためのX68000/マシン語プログラミング
ハード工作/ショートプロ/ANOTHER CG WORLD/Z80's Bar
吾輩はX68000である/Computer Music入門

●新製品紹介　Z'sSTAFF PRO-68K ver.3.0
LIVE in '92　Shake the Street/Ancient relics
THE SOFTOUCH　スピンディジーII/ロイヤルブラッド/ライフ&デス他
全機種共通システム　実践Small-C講座(3)COMMAND.OBJ2

7月号
特集　超空間美術論
特別付録　DōGA CGAシステム＆お試しディスク(5"2HD)

連載　よいこのSX-WINDOW/響子 in CGわ～るど/Z80's Bar
ANOTHER CG WORLD/大人のためのX68000
Computer Music入門/ハード工作/ショートプロ

●試用レポート　V70アクセラレータボード
LIVE in '92　Bye Bye My Love/MATERIAL GIRL/ヴェクザシオン
THE SOFTOUCH 将棋聖天&棋太平68K/シムアース/太閤立志伝
全機種共通システム　実践Small-C講座(4)関数リファレンス

8月号
特集　プログラミング再入門

連載　響子 in CGわ～るど/吾輩はX68000である/よいこのSX-WINDOW
マシン語プログラミング/ハード工作/ANOTHER CG WORLD
大人のためのX68000/Computer Music入門/ショートプロ

●新製品紹介　MATIER/TG100/SOUND SX-68K
LIVE in '92　氷穴/ガラガラヘビがやってくる/風の贈り物
THE SOFTOUCH　三國志III/シムアース/ウルティマVI/バトルテック
全機種共通システム　実践Small-C講座(5)ワイルドカード
グラフィックライブラリGRAPH.LIB

9月号
特集　数値演算の熱い逆襲

連載　DōGA CGアニメーション講座/大人のためのX68000
響子 in CGわ～るど/吾輩はX68000である/ショートプロ
マシン語プログラミング/ハード工作/ANOTHER CG WORLD

●新製品紹介　MATIER/MIREGE System Model Stuff
LIVE in '92　恋をしようよ Yeah! Yeah!/ゆめいっぱい
THE SOFTOUCH　ファイナルファイト/ライジングサン/
ヨーロッパ戦線/シューティング68K GAMES
全機種共通システム　O-EDIT & MODCNV

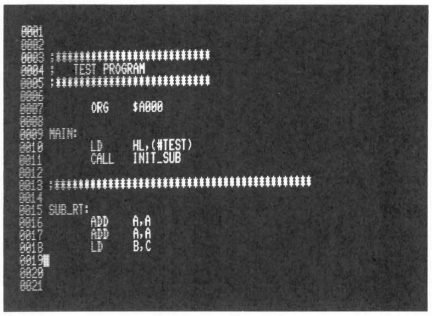
10月号
特集　DTMへの招待

連載　DōGA CGアニメーション講座/大人のためのX68000
響子 in CGわ～るど/吾輩はX68000である/ショートプロ
マシン語プログラミング/ハード工作/ANOTHER CG WORLD

●試用レポート　X68000用 CD-ROM ドライブ
LIVE in '92　美少女戦士セーラームーン/笑顔を探して 他
THE SOFTOUCH　ポピュラスII/リーディングカンパニー/
ネクタリス/サークII
全機種共通システム 実践Small-C講座(6)SLENDER HUL

THE SENTINEL

〈対応機種一覧〉●MZ-80K/C/700/1500●MZ-80B/2000●MZ-2500/2861●X1●X1turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA 7●FM-7/77/AV●PC-286/386/9801/98●X68000
掲載されたプログラムの利用には各機種用のS-OS"SWORD"システムが必要です。



第126部　EDIT　実践Small-C講座(7)

●EDIT

今月は，久しぶりにテキストエディタが登場します。コンパクトにまとまったカーソルエディタで，「REDA」に内蔵されていたエディタを拡張したものです。

このエディタの大きな特徴は，ラインエディタながらも，エディット画面でコントロールコマンドを書き込むことができる点です。コマンドを書き込む部分は，コントロールカラムと呼ばれる，行番号と編集テキストの間にある空白部分。ページ切り替えや行挿入，行マークによる削除コマンドなどがあり，コマンドモードに戻って操作することなく，エディット画面のみで快適な操作が可能となったのです。慣れれば，カーソルエディタながらもスクリーンエディタ並みの操作性を実現でき，ずいぶん使い込んでいる人もいるようです。

さて，どのへんが拡張されたかというと，ファイルアクセス部分に拡張が施されました。編集したテキストファイル名を“UPDATE.$$$”というテンポラリファイルに残すようにしたのです。

なぜ，このようなファイルをわざわざ残すようにしたかというと……理由は本文にあるとおりMAKEプログラムを使うためです。MAKEプログラムを使うにあたって，テキスト更新情報として必要なタイムスタンプ情報を，別ファイルへ更新したファイル名を登録することで解決したのです。

●MAKEへの布石

MAKEを効率よく使うために拡張機能を備えた「EDIT」。しかし，裏を返せばMAKEプログラムを使うためには，いままで使ってきたエディタではなく「EDIT」を使わなくてはならない，ということにほかなりません。ところが，エディタというのは手に馴染むもの。いきなり使い慣れたものを変えさせるのは無理があります。

そこで，「EDIT」で採用したファイル拡張部分のプログラムを募集します。

必要なサブルーチンは3つ。仕様は以下に書かれたものを参考にしてください。

1)　INIT_UPDATE

起動時に編集したいテキストファイル名を指定できるようにする。

例)　#　E-MATE TEST.S

また，エディタ起動時に編集したテキスト情報“UPDATE.$$$”があれば，メモリに確保したワークに読み込む（512バイト程度，特殊ワークに確保することは不可)。さらに起動時に設定したファイルが，“UPDATE.$$$”に存在するかチェック。

2)　UPDATE_SET

新規テキストを読み込んだ場合に“UPDATE.$$$”の内容を更新する。テキストのアペンドと区別するために“UPDATE　OK(Y/N)?”と聞いて“Y”が押されたときのみワークを更新するのが望ましい。

3)　QUIT_UPDATE

エディタ終了時に，更新した“UPDATE.$$$”の内容を出力する（新規作成する)。

プログラムは最終出力が同じなら，どのようなものでもかまいません。それほどむずかしいものではありませんので，ぜひチャレンジしてください。

●S-OSの系譜(38)

1989年12月号では，SLANG用リダイレクションライブラリ「DIO.LIB」が登場しました。このライブラリを使ってSLANGでプログラムを組めば，自動的にリダイレクト，パイプ機能をサポートしたプログラムを作成できるようにしたものです。

基本的に“すべてがファイル”という概念を持ったUNIXなどの世界では，コマンドからプログラムが実行されると，標準入力(主にキーボード)からデータを入力し，標準出力（主に画面）へ出力します（リダイレクトとは，標準入力，標準出力を変えることをいいます)。

ちょっとしたデータ出力，加工なども，リダイレクトすることで，結果をいろいろなデバイスに出力でき，さらにパイプラインを設置することで，出力したデータを新しい入力データにする。これらが簡単なオペレーションだけで実現できるのです。

1992■インデックス

■92年1月号
第115部　LINER
■92年2月号
第116部　シミュレーションゲームPOLANYI
■92年3月号
第117部　カードゲームKLONDIKE
■92年4月号
第118部　オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部　COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部　COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部　関数リファレンス実践Small-C講座(4)
■92年8月号
第122部　ワイルドカード実践Small-C講座(5)
第123部　グラフィックライブラリ GRAPH.LIB
■92年9月号
第124部　O-EDIT&MODCNV
■92年10月号
第125部　SLENDER HUL実践Small--C講座(6)

THE SENTINEL

全機種共通
S-OS“SWORD”要

EDIT

実践Small-C講座(7)

Tatsuya Ishigami
石上 達也

編集したテキスト名を出力する機能を追加した，カーソルエディタ「EDIT」が登場。来月から制作の始まるMAKEプログラムを動かすために必要な機能ですから，ぜひ入力しておきましょう。

Small-Cの環境整備を行ってきたこのシリーズもそろそろ佳境を迎えます。てなわけで，今回はエディタを作ってみます。

さて，エディタを作るにしても，どのようなタイプのエディタを作ることにしましょうか。まず，それで悩みました。重いスクリーンエディタもいいけど，手軽に入力できて小回りの利くカーソルエディタも捨てがたい。今回は，ある機能を付けるため新しくエディタを作ることが目標でした。やっぱり，使い慣れたものをベースにして必要な機能を付加しよう。そんなことを考えながらこの「EDIT」を作りました。

エディタについて

いまさらいうまでもなくSmall-Cはコンパイラです。コンパイラというのは，人間にわかりやすい言語（この場合はC言語）で書かれたソースプログラムを，コンピュータが実行できる言語（最終的には0と1の信号レベル）へと変換します。コンパイラの役割はただこれだけです。純粋にコンパイラといったら，これ以上のことはしませんし，これ以下のものでもありません。

そして，コンピュータが実行できる言語に変換されたプログラムを実行するのは，もちろんコンピュータです。これには問題ありませんね。さて，反対側の人間にわかりやすい言語というのは，どうやって作るのでしょう。

もちろん，人間がウンウンうなって作るのですが，その結果をコンピュータに教え込まなくてはなりません。これがBASICやエディタアセンブラならば，その機能を使ってウンウン考えた結果を入力してやればよいのですが，あいにくSmall-CやWZDにはそのような機能は備わっていません。

備わっていないものは，使うことをあきらめるか代わりのものを用意するしかありません。いままでは，E-MATEとかREDAのエディタ部分を使用していたのですが，そろそろ別のエディタもあってもいいのではないか，というわけで今回のEDITです。

REDAについて

エディタというのは，非常に多様な場面で使われるわけで，ひょっとしたら作った本人が思いもつかないような使われ方をされる場合があります。たとえば，セーブ前に暴走して打ち込み途中のテキストが全部パーになってしまった，なんていうことは絶対に避けなければなりません。エラーチェックなどもしっかりとやっておく必要があります。以前，満開製作所社長の祝氏もエディタを作るときは，

「きれいなプログラムにできるとは金輪際思わないこと」

「エディタを作るということは単なる力仕事なのである」

などとおっしゃっていました。だいたい頭の中にプログラムの大まかな骨組みは浮かぶのですが，実際にコーディングを始めると「うんがー」となってしまいそうで，怖くて始められません。

自分しか使わないプログラムであればそれでもなんとかなるのでしょうが，今回のプログラムは雑誌に載ることが大前提であるし，しかも連載という形をとっていて期日までに仕上げる必要がある。このような理由からOh!X1989年2月号に掲載された「高速エディタアセンブラREDA」のエディタ部分を叩き台にして，それをシェイプアップしたものを掲載させていただきます。

で，手直し部分はちょっとのはずでしたが，あっちに手を入れたらこっちにも手を入れなければ……というように結局ほとんど別のプログラムになってしまいました(エンバグしてなければいいのですが)。そんな調子で，不要と思われる機能もいくつか削ってしまいました（どの機能がないかは後述します)。例によってソースリストが「ここ」にありますので，必要な機能は各自で追加してください。非常にコンパクトにまとまったエディタです。

使い方

ここからは，5月号のCOMMAND.OBJが走っている状態で話を進めていきます。

まず，MACINTO-Cなどのツールを使って，リスト1のプログラムをAドライブに“EDIT”というファイルネームでセーブします。起動方法は，

A>EDIT　ファイル名　[オプションスイッチ]

と入力します。ファイル名で示されたファイルがカレントディスク上に存在すれば，その中身を読み込んでからエディタは立ち上がりますし，存在していなければ，なにも入力されていない状態で立ち上がります。基本的な操作方法は，REDAのエディタと同じです。ただし，REDAでいうところのホットスタート機能はありませんので，Qコマンドで抜け出すとそのテキストの内容は破棄されてしまいます。一度破棄したテキストを復活する機能も用意されていませ

ん。注意してください。

今回のエディタはREDAのエディタ部分に対して若干の拡張機能がついています。若干のものですが，これがあるとじゃまだというときには，起動時のコマンドラインの［オプションスイッチ］のところに“/N”を指定してください。スイッチを指定することにより，ファイル“UPDATE.$$$”をアクセスすることはなくなります。オプションスイッチは“/N”だけです。

起動するとなにかしらのファイルを読み込んだ場合にはその先頭行が，そうでない場合には左上端に数字の0001が表示されます。カーソルは左下角で点滅しているはずです。この状態をコマンドモードといい，表1に示すようなコマンドが使えます。コマンドモードでは，主にテキストの検索，変更などを行います。この状態からEコマンドで，エディットモードへ移行することができます。エディットモードでは表2に示すような機能が使えます。エディットモードでは，主にテキストの変更，削除，挿入など実際のテキスト編集を行います。そして，エディットモードからコマンドモードに戻るには，SHIFT＋BREAKキーを一緒に押します。

表1にあるコマンド一覧表は，一部ホットスタート関連のコマンドがないだけでほとんどREDAと同じです。同じですが，REDAを使っていなかった人のために，もう一度操作法を復習しておきましょう。

コマンドモード

●指定した行からの表示

コマンドモードから“G”を入力すると，表示を開始する行番号をたずねてくるので，10進数でその行番号を入力してください。存在しない行を入力するとテキストの最後（つまりは真っ白な部分）が表示されてしまいますので，“Z”コマンドで1ページ前に戻してください。

●ファイル入出力

コマンドモードから“F”を入力すると，「Load, Save, Get-cut, Cut-save, Dir」と質問してきます。それぞれ，

Load：新しい別のテキストを読み込む
Save：現在編集中のテキストを保存する
Get-cut：指定したASCIIファイルをカットバッファへ読み込む
Cut-save：カットバッファの内容を指定したファイルへ書き出す
Dir：指定したデバイスのディレクトリ情報を表示する

となっているので，行いたい作業を最初の1文字（メニュー名の大文字の部分）で答えてください。一度指定されたファイル名は，次回以降の作業でも保存されます。

●文字列の検索/置換

文字列の検索や置換を実行すると，まず「Begin, Here」と質問してきます。作業をテキストの先頭から行うか，それとも現在表示されているページから行うかの選択です。“B”(小文字も可)以外の文字が入力されると，後者の状態で検索が実行されます。

検索を実行すると，画面には検索文字列のあるページが表示され，その文字列の先頭でカーソルが点滅しています。このページで作業を行いたい場合には，SHIFT＋BREAKキーを押し検索を中止します。すると，エディットモードになるので，このモードで作業を行ってください。

同様に置換を実行すると，画面に検索文字列のあるページが表示され，その文字列の先頭でカーソルが点滅しています。このページで作業を行いたい場合には，SHIFT＋BREAKで，置換を中止できるのは検索と同じです。ここで，リターンキーを押すと文字列の置換が行われ，次の検索文字列のページが表示されます。置換を行いたくないときには，ここでリターンキー以外のキーを押します。

エディットモード

テキストを表示すると必ず，画面の左端に4桁の10進数が表示されています。そして，そのあとに1文字分の空白があり，編集しているテキストの内容がこれに続いています。この1文字分の空白のことをEDITでは（ということはREDAでもそうなのですが）コントロールカラムといいます。EDITのエディットモードでは，このコントロールカラムのところにコマンドを書き込み，リターンキーを押すことによってコマンドを実行します。

●行の挿入

まず，テキストを入力するには，テキストが入力される行を確保しなければいけません。そこで使うのが“＋”コマンドです。このコマンドでは，現在行の次から10行の空白行を挿入します。まず，このコマンドで空白行を確保しておいてからテキストの入力を行ってください。

●カット＆ペースト

EDITではコピー機能はサポートされていません。コピーを行いたいときには，以下のように作業を行います。

まず，“.”コマンドで転送したい領域の一端を押さえます（このことを「行をマーク」するといいます）。この一端というのは転送したい領域の開始行でも終了行でもかまいません。そして，“<”コマンドをもう

図1　画面構成

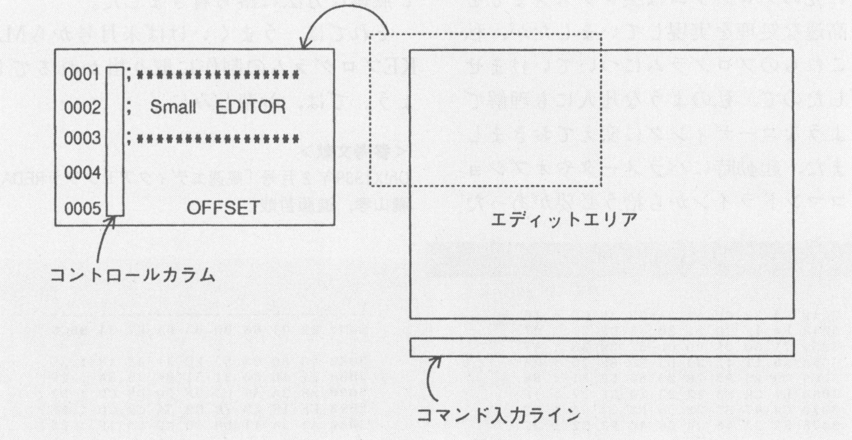


表1　コマンドモード

E：エディットモードに入る。カーソルは画面最上行の入力開始位置に移動
V：次のページを表示
Z：前のページを表示
G：表示を開始する行番号を入力
S：検索（順方向のみ）
C：置換（順方向のみ）
F：ファイルのロード/セーブ

表2　エディットモード

＋：現在行の後ろに空白行を10行挿入する
—：現在行を削除する
/：現在行と，続く改行だけの行を削除する
.：現在行をマークする
,：現在行とマーク行を交換する
[：現在行を最上行として表示する
=：現在行を中心行として表示する
]：現在行の前の行を最下行として表示する(前ページを表示する)
)：後ろのページを表示する
<：マーク行から現在行までを削除する
>：削除バッファの内容を現在行の前に挿入する

一端のところで使用することによって，この“.”と“<”で囲まれた領域をカットバッファに転送することができます。このままでは，指定領域の削除にすぎません（ということは，指定領域の削除にはこれを使うということ)。“>”コマンドを使っていまカットバッファに転送した内容を元の位置に復元します。これで，テキストには元どおりになったハズです。そして，カットバッファにはコピーしたい内容が転送されているハズです。この内容をコピーしたい行の先頭で，もう一度“>”コマンドを使用すれば，この内容が転送されます。これでコピー作業が代用できるのです。

なにが違うのか?

しかし，REDAのエディタ部分をただ抜き出して掲載したのでは面白くありません。今回は機能的に少しばかり手を加えてあります。エディタの骨格部分ではなく，ファイル関係のちょっとした部分にです。

テキストをセーブするとき，カレントディスク上に“UPDATE.$$$”というファイルに編集したテキストのファイルネームを登録します。カレントディスク上に“UPDATE.$$$”というファイルがないときには，新たに作って登録します。この“UPDATE.$$$”というファイルはASCIIファイルなので，エディタで読み込んで内容を表示させたり更新したりすることができます。

このようにすると，なにかいいことがあるのでしょうか?

ここで，Small-C本体のような複数に分割されたプログラムを作成する場合を思い出してください。エディタを使っていくつかのプログラムを変更し，コンパイルし直そうかと思ったとき，あなたは最初に変更したプログラムを覚えているでしょうか?

1時間も2時間も前に，ほんのちょこっと変更した最初のファイルの名前を覚えていられますか。そして，それらの変更がいったいどのようにどのファイルに影響を及ぼすのか。これらを正確に把握していられるでしょうか。

そんな面倒臭く，かつ単純な作業はコンピュータにやらせてしまったほうが利口というものです。そんなことはわかっていたのですが，いままで“SWORD”ではそういう試みはなされてきませんでした。それはMS-DOSやHuman68kでいうところのタイムスタンプの機能がなかったからです。

つまり，MS-DOSなどではこのタイムスタンプの機能を使って，テキストファイルのセーブされた（あるいはセーブし直された，更新された）時刻を知ることができます。こうして，コンピュータは「おや，このファイルはオブジェクトファイルよりもあとにセーブされているな。ということは，このファイルはもう1回コンパイルしなきゃいけないんだな」ということを自動的に判断し，自動的に実行してくれます。

コンピュータがそのようなことをやってくれれば，人間は問題のありそうな部分のソースプログラムをいじることに，より集中できるようになります。

そんなわけで，今回このような機能をつけました。そして，なるべく近いうちにそのコンパイル作業などを自動的にコーディネートしてくれるようなプログラムを発表したいと思っています。このようなプログラムは，MS-DOSやHuman68kの世界ではMAKEプログラムと呼ばれています。

プログラムについて

先ほども述べたようにこのプログラムは，REDAからほとんどもらってきたプログラムです。ほとんどの部分に手が入っていますが，基本的には借りものです。借りものでありながら「プログラムについて」もないのですが，少しばかりお話を。

まず，「エディタアセンブラ」のアセンブラ部分をすっぽり抜き取っていますので，そのぶんフリーエリアは大きいです。そして，タブコードのあるテキストには対応していません。

次に元のプログラムは裏レジスタまでを使い高速な処理を実現していましたが，私にはこれらのプログラムについていけませんでしたので，私のような凡人にも理解できるようなコーディングに変えておきました。また，起動時にパラメータやオプションをコマンドラインから拾う必要があったので，プログラムの最初のほうはかなり手を入れています。このEDITは5月号のCOMMAND.OBJから使うことを想定しているので，ホットスタート機能を省略しました。

REDAはQコマンドで「さくっ」とエディタモードから抜け出せてしまいました。EDITはコマンドラインからパラメータつきで起動する関係上，ホットスタート機能というものがありません。そうなるとあまり簡単に抜け出せては困ります。そこで，削除や挿入や更新を行ったテキストを抱えたままEDITを抜け出そうとすると，

Are you sure?

と聞いてきて，これに対して「Y」または「y」が入力されないとコマンドラインへ抜け出せないようになっています。そのため，テキストに対しなにか操作を行ったときには，そのことを示すフラグ（リスト2中のラベルIsTxtChg)を1にするよう，各所にプログラムを埋め込んでおきました。

REDAではカットバッファをテキストのすぐ後ろに置いていましたが，EDITでは特殊ワークエリアにとってあります。そのため，多少フリーエリアが広くなりました。またREDAでは，新しくテキストをロードするとカットバッファの内容も一緒に消えてしまいましたが，EDITでは保存されるようになっています。

* * *

今回発表したEDITでは，タイムスタンプ機能の代用として，編集したファイル名を登録するテンポラリファイルを自動生成する機能を追加しました。これとは別に，ディスクの未使用領域をタイムスタンプフラグとして使用することも考えられますが，この方法では一部の機種しかサポートできないうえに，現在の“SWORD”の規格外使用となってしまいます。なるべく標準的な使い方をしなくてはならない，ということで無難な方法に落ち着きました。

それでは，うまくいけば来月号からMAKEプログラムの制作に取り掛かれるでしょう。では，お楽しみに。

<参考文献>
Oh!X1989年2月号「高速エディタアセンブラREDA」瀧山孝，進藤哲哉

リスト1

```
3000 ED 73 B9 3A ED 7B 6A 1F : 44
3008 CD F0 30 AF 32 B8 3A 21 : E1
3010 00 00 22 B6 3A 21 00 40 : 73
3018 22 AD 3A 36 0D 23 36 00 : A5
3020 22 AB 3A 21 01 00 22 AF : FA
3028 3A 22 E3 37 3A 66 35 A7 : F2
3030 28 0D 3E 04 11 66 35 CD : F0
3038 A3 1F CD E2 33 18 03 CD : 8C

3040 F3 32 CD A1 3A 3E 3E CD : 16
3048 F4 1F CD 21 20 E6 DF 21 : 07
3050 71 30 06 0D CD 5E 30 38 : 47
3058 E9 11 42 30 D5 E9 BE 20 : 08
3060 09 23 F5 7E 23 66 6F F1 : 88
3068 B7 C9 23 23 23 10 EF 37 : 1F
3070 C9 47 BC 32 56 E2 32 5A : C2
3078 55 33 45 CB 36 4C F3 32 : 3F

------------------------------------
SUM: 22 01 68 B0 B3 6A F7 6A 96C6

3080 46 B0 33 53 ED 31 43 48 : 25
3088 31 4D 30 31 51 8F 30 3A : 29
3090 B8 3A A7 C4 9F 30 D8 CD : D1
3098 EE 1F ED 7B B9 3A C9 CD : FE
30A0 A1 3A 11 D8 30 CD E5 1F : C5
```

```
30A8 CD A1 3A 11 C4 30 CD E5 : 5F
30B0 1F CD 21 20 E6 DF FE 59 : 49
30B8 C8 CD A1 3A 11 D8 30 CD : 56
30C0 E5 1F 37 C9 41 72 65 20 : 3C
30C8 59 6F 75 20 53 75 72 65 : FC
30D0 20 3F 28 59 2F 4E 29 00 : 86
30D8 20 20 20 20 20 20 20 20 : 00
30E0 20 20 20 20 20 20 20 20 : 00
30E8 20 20 20 20 20 20 20 00 : E0
30F0 AF 32 66 35 3E 01 32 B5 : A2
30F8 3A 2A 76 1F 7E 23 A7 C8 : 09
--------------------------------
SUM: 19 54 14 FC 60 97 2D 88 5287

3100 FE 20 20 F8 FE 2F 28 1A : A5
3108 FE 2D 28 16 06 13 11 66 : F9
3110 35 7E 12 A7 28 08 FE 20 : BA
3118 28 04 13 23 18 F3 AF 12 : 2E
3120 18 DA 7E E6 DF FE 4E 20 : A1
3128 D3 23 AF 32 B5 3A 18 CC : AA
3130 CD A1 3A CD F1 1F 21 00 : A6
3138 40 CD BE 1F 3E 2D CD F4 : 16
3140 1F 2A AB 3A CD BE 1F C9 : A1
3148 CD 41 32 FE 1B C8 11 CF : 01
3150 31 CD 56 3A CD 87 32 C8 : DC
3158 1A B7 C8 21 D6 31 CD 69 : F7
3160 32 21 A5 32 CD 9E 33 E5 : AD
3168 21 D6 31 CD 9E 33 D1 B7 : 4E
3170 ED 52 22 EB 31 CD 03 32 : 7F
3178 38 0B CD 21 20 FE 0D 28 : 84
--------------------------------
SUM: 00 7D 52 7A 4E 9B 7D 51 BB08

3180 08 FE 1B 20 F0 CD F3 32 : 23
3188 C9 3E 01 32 B8 3A 2A EB : 41
3190 31 7C B7 F4 BD 39 DA 1D : 45
3198 35 2A EB 31 7C B5 28 1E : F2
31A0 E5 7C E6 80 20 0D 2A BA : D8
31A8 32 2B 22 D9 39 E1 CD 5C : 9B
31B0 37 18 0B 2A BA 32 2B 22 : BD
31B8 D9 39 E1 CD 77 37 21 D6 : 65
31C0 31 ED 5B BA 32 1B 7E B7 : B5
31C8 28 AB 23 12 13 18 F7 52 : 7C
31D0 65 70 6C 63 65 3A 20 20 : 83
31D8 20 20 20 20 20 20 20 20 : 00
31E0 20 20 20 20 20 20 20 20 : 00
31E8 20 20 00 00 00 CD 41 32 : 80
31F0 FE 1B C8 CD 03 32 D8 CD : 88
31F8 21 20 FE 0D 28 F5 FE 1B : 82
--------------------------------
SUM: 9B 7D A2 10 80 ED 4E E9 5330

3200 20 F5 C9 2A AB 3A ED 5B : 35
3208 BA 32 B7 ED 52 7D B4 37 : 4A
3210 C8 44 4D EB 11 A5 32 1A : 46
3218 ED B1 37 C0 22 BA 32 2B : CE
3220 23 13 1A B7 28 05 BE 28 : 1A
3228 F7 18 D8 2A BA 32 2B CD : F5
3230 AD 36 22 B7 37 CD 4B 39 : 44
3238 2A BC 37 CD 1E 20 B7 C9 : A8
3240 C9 CD A1 3A 11 91 32 CD : 12
3248 E5 1F CD 21 20 E6 DF FE : D5
3250 42 2A AD 3A 20 03 21 00 : 97
3258 40 22 BA 32 11 9E 32 CD : FC
3260 56 3A CD 87 32 C8 21 A5 : A4
3268 32 06 14 1A 13 77 B7 C8 : 6F
3270 FE 5E 28 04 23 10 F4 C9 : 78
3278 1A 13 FE 5E 28 04 E6 DF : 7A
--------------------------------
SUM: 50 22 2B F1 59 A5 06 7B D5F3

3280 D6 40 77 23 10 E5 C9 D5 : 43
3288 ED 5B 76 1F 1A D1 FE 1B : E1
3290 C9 42 65 67 69 6E 2C 20 : FA
3298 48 65 72 65 20 00 53 65 : 5C
32A0 61 72 63 68 3A 20 20 20 : 38
32A8 20 20 20 20 20 20 20 20 : 00
32B0 20 20 20 20 20 20 20 20 : 00
32B8 20 00 00 00 11 D5 32 CD : 05
32C0 56 3A CD 87 32 C8 CD 22 : DD
32C8 3A 22 AF 3A CD 15 3A 22 : 83
32D0 AD 3A C3 F3 32 47 6F 74 : F9
32D8 6F 3A 00 FE 30 D8 FE 3A : E7
32E0 3F C9 ED 5B B1 3A 1A B7 : 0C
32E8 C8 ED 53 AD 3A 2A B3 3A : 06
32F0 22 AF 3A AF 32 54 33 2A : 9D
32F8 AF 3A 22 B3 3A 2A AD 3A : 09
--------------------------------
SUM: 19 63 42 D2 F6 37 F9 F9 523E

3300 22 B1 3A 3E 0C CD F4 1F : 37
3308 2A B1 3A 7E A7 C8 CD 3D : 0C
3310 33 3A 54 33 21 5B 1F BE : 4D
3318 D0 2A B3 3A 23 22 B3 3A : 19
3320 2B CD DB 39 CD F1 1F ED : D6
3328 5B B1 3A CD E8 1F 1A 13 : 47
3330 FE 0D 20 FA ED 53 B1 3A : 50
3338 CD EE 1F 18 CB E5 CD 9E : 0D
3340 33 3A 5C 1F 67 7D F5 3A : FB
3348 54 33 3C 32 54 33 F1 94 : 01
3350 30 F4 E1 C9 1B ED 5B AF : E0
3358 3A 1B 7A B3 C8 AF 32 54 : 7F
3360 33 2A AD 3A 2B 22 AD 3A : 78
3368 CD 95 33 E5 CD 3D 33 2A : F1
3370 54 33 21 5B 1F BE E1 38 : F9
3378 EC 2A AD 3A 11 00 40 37 : 85
--------------------------------
SUM: D1 D7 70 C2 2A C3 BE E0 5D9A

3380 ED 52 30 04 ED 53 AD 3A : 9A
3388 2A AD 3A CD AD 36 22 AF : 92
3390 3A CD F3 32 C9 2B 2B 7E : C9
3398 FE 0D 20 FA 23 C9 C5 01 : D7
33A0 FF FF 7E 23 03 FE 0D 28 : D5
33A8 03 A7 20 F6 60 69 C1 C9 : 13
33B0 11 3D 35 CD E5 1F CD 21 : 42
33B8 20 E6 DF FE 4C 28 17 FE : 6C
33C0 53 CA 65 34 FE 43 CA 98 : 59
33C8 34 FE 47 CA 17 34 FE 44 : D0
33D0 CA 7A 35 C3 F3 32 3A B8 : 53
33D8 3A A7 C4 9F 30 D8 CD F9 : 12
33E0 34 D8 CD 09 20 DA 34 35 : 45
33E8 28 08 CD 9D 1F CD EE 1F : 93
33F0 18 E4 21 01 00 22 AF 3A : 29
33F8 22 E3 37 21 00 40 22 AD : 6C
--------------------------------
SUM: A3 32 C6 09 91 B5 33 40 6FE4

3400 3A 22 70 1F ED 5B 72 1F : C4
3408 19 2B 22 AB 3A CD A6 1F : DD
3410 DA 34 35 CD F3 32 C9 CD : CB
3418 F9 34 D8 CD 09 20 DA 34 : 09
3420 35 2A AB 3A ED 5B 72 1F : 1D
3428 19 DA 1D 35 EB 2A 6A 1F : E3
3430 B7 ED 52 DA 1D 35 2A AB : F7
3438 3A 23 22 70 1F E5 CD A6 : 66
3440 1F E1 DA 34 35 ED 4B 72 : ED
3448 1F 0B ED 43 B6 3A ED 5B : 92
3450 AB 3A 13 21 00 00 1A 13 : 46
3458 CD 9A 1F 23 0B 78 B1 20 : FD
3460 F5 CD F3 32 C9 CD F9 34 : AA
3468 D8 2A AB 3A 11 00 40 B7 : EF
3470 ED 52 22 72 1F 21 00 00 : 13
3478 22 70 1F 22 6E 1F CD AF : DC
--------------------------------
SUM: F7 42 B3 D8 94 C5 97 68 81F5

3480 1F DA 34 35 21 00 40 22 : E5
3488 70 1F CD AC 1F DA 34 35 : 6A
3490 CD D7 35 AF 32 B8 3A C9 : 75
3498 2A B6 3A 7C B5 CA C4 1F : F8
34A0 CD F9 34 D8 ED 4B B6 3A : FA
34A8 03 11 00 40 21 00 00 CD : 42
34B0 E7 34 1B AF 12 2A B6 3A : 11
34B8 23 22 72 1F 21 00 00 22 : 19
34C0 70 1F 22 6E 1F CD AF 1F : D9
34C8 38 6A 21 00 40 22 70 1F : B4
34D0 CD AC 1F 38 5F CD D7 35 : 08
34D8 ED 4B B6 3A 03 11 00 40 : 7C
34E0 21 00 00 CD E7 34 C9 1A : EC
34E8 F5 CD 94 1F 12 F1 CD 9A : DF
34F0 1F 13 23 0B 78 B1 20 EF : 98
34F8 C9 11 61 35 CD 56 3A CD : 9A
--------------------------------
SUM: C0 57 61 FE 67 CA C4 C5 9AC2

3500 87 32 37 C8 01 13 00 11 : DD
3508 05 00 2A 76 1F 19 11 66 : 54
3510 35 ED B0 3E 04 11 66 35 : C0
3518 CD A3 1F B7 C9 CD C4 1F : BF
3520 CD A1 3A 11 29 35 C3 E5 : BF
3528 1F 20 4E 4F 20 4D 45 4D : DB
3530 4F 52 59 00 CD 33 20 CD : E7
3538 21 20 C3 F3 32 4C 6F 61 : 45
3540 64 2C 20 53 61 76 65 2C : 6B
3548 20 47 65 74 2D 63 75 74 : B9
3550 2C 20 43 75 74 2D 73 61 : 79
3558 76 65 2C 20 44 69 72 20 : 66
3560 00 46 49 4C 45 3A 4E 4F : F7
3568 4E 41 4D 45 20 20 20 20 : A1
3570 20 20 20 20 20 20 20 20 : 00
3578 20 00 11 CF 35 CD E5 1F : 06
--------------------------------
SUM: 9E 94 8F 62 35 C1 04 FA D1FD

3580 CD 21 20 FE 1B C8 FE 0D : FA
3588 CC 24 20 32 5D 1F CD EE : 79
3590 1F CD 06 20 11 B2 35 CD : D7
3598 E5 1F ED 5B 76 1F CD D3 : 81
35A0 1F 1A FE 1B C8 21 05 00 : 40
35A8 19 11 66 35 01 13 00 ED : C6
35B0 B0 C9 48 69 74 20 43 52 : 53
35B8 20 6F 6E 20 66 69 6C 65 : BD
35C0 6E 61 6D 65 20 6F 72 20 : C2
35C8 42 52 45 41 4B 0D 00 0D : 7F
35D0 44 72 69 76 65 3A 00 3A : 6E
35D8 B5 3A A7 C8 3E 04 11 A2 : 53
35E0 35 CD A3 1F CD 09 20 38 : F3
35E8 5A 2A 72 1F ED 5B AB 3A : 42
35F0 19 B7 ED 72 D2 1D 35 13 : 66
35F8 ED 53 70 1F CD A6 1F DA : 3B
--------------------------------
SUM: E4 F4 81 37 09 56 23 A7 55AA

3600 70 36 ED 5B AB 3A 2A 72 : 6F
3608 1F 23 19 11 66 35 1A 13 : 34
3610 FE 20 28 FA 77 23 A7 20 : A1
3618 F5 77 2B 36 0D 23 ED 5B : 45
3620 AB 3A B7 ED 52 23 22 72 : 92
3628 1F 21 00 00 22 70 1F 22 : 13
3630 6E 1F CD AF 1F 38 39 2A : C3
3638 AB 3A 22 70 1F CD AC 1F : 2E
3640 38 2E C9 21 66 35 CD 77 : 2F
3648 36 CD 9E 33 23 22 72 1F : AA
3650 21 00 00 22 70 1F 22 6E : 62
3658 1F CD AF 1F 38 12 21 66 : 8B
3660 35 22 70 1F CD AC 1F 38 : B6
3668 07 21 6B 35 CD 82 36 C9 : 16
3670 11 8E 36 CD E5 1F C9 E5 : 54
3678 7E 23 A7 20 FB 2B 36 00 : D1
--------------------------------
SUM: DE 60 CD 7E F2 4D D4 3A F505

3680 E1 C9 E5 7E 23 FE 0D 20 : 5B
3688 FA 2B 36 00 E1 C9 46 49 : 94
3690 4C 45 20 41 43 43 45 53 : 10
3698 53 20 45 52 52 4F 52 20 : 1D
36A0 4F 4E 55 50 44 41 54 45 : 60
36A8 2E 24 24 24 00 C5 D5 01 : 35
36B0 00 00 11 00 40 1A A7 28 : 3A
36B8 0D 13 FE 0D 20 F7 03 E5 : 2A
36C0 B7 ED 52 E1 30 EF 60 69 : BF
36C8 D1 C1 C9 21 04 00 22 BC : 5E
36D0 37 2A BC 37 CD 1E 20 ED : 4C
36D8 5B 76 1F CD D3 1F CD 18 : 94
36E0 20 2E 04 22 BC 37 1A FE : 7F
36E8 1B CA A1 3A CD DB 32 38 : D2
36F0 E0 3E 01 32 B8 3A CD 32 : 42
36F8 3A 22 B7 37 32 BB 37 ED : 5B
--------------------------------
SUM: 73 84 5B 5D 84 A3 7C AE 1FB9

3700 53 B9 37 CD 15 3A 22 D9 : 5A
3708 39 21 D1 36 E5 3A BB 37 : 72
3710 06 0A 21 BE 37 CD 5E 30 :.81
3718 38 01 E9 2A D9 39 CD 9E : C9
3720 33 E5 2A B9 37 CD 9E 33 : D0
3728 3A BB 37 FE 2B 20 04 11 : 8A
3730 0A 00 19 D1 B7 ED 52 7D : 67
3738 B4 28 54 EB 2A B1 3A 19 : 49
3740 22 B1 3A EB 7C E6 80 C2 : 9C
3748 55 37 CD BD 39 D8 CD 5C : 50
3750 37 CD 8F 37 C9 CD 77 37 : 0E
3758 CD 8F 37 C9 ED 5B AB 3A : 89
3760 D5 19 22 AB 3A E1 ED 4B : 0E
3768 D9 39 B7 ED 42 23 44 4D : AC
3770 2A AB 3A EB ED B8 C9 EB : 53
3778 2A D9 39 B7 ED 52 ED 5B : 7A
--------------------------------
SUM: 72 C7 F9 40 0E F9 8C 25 9BEB

3780 D9 39 7E 23 12 13 A7 20 : 9F
3788 F9 1B ED 53 AB 3A C9 2A : 2C
3790 D9 39 CD D0 39 28 09 1A : 33
3798 B7 28 05 77 23 13 18 F7 : A0
37A0 36 0D 23 3A BB 37 FE 2E : BE
37A8 28 32 FE 2B C0 06 0A 36 : 89
37B0 0D 23 10 FB C3 F3 32 00 : 23
37B8 00 00 00 38 00 00 2D 32 : 97
37C0 38 2F 41 38 3E 6C 38 3C : FE
37C8 B6 38 3D 28 39 28 22 38 : 0E
37D0 29 27 38 2C E5 37 5B F3 : 1E
37D8 37 5D 04 38 2A B7 37 22 : 0A
37E0 E3 37 C9 00 00 2A E3 37 : 27
37E8 ED 5B B7 37 ED 53 E3 37 : 90
37F0 C3 2B 39 2A B7 37 22 AF : 10
37F8 3A 2A D9 39 22 AD 3A CD : 4C
--------------------------------
SUM: E8 E9 BA B3 A3 9B 06 64 EE13

3800 F3 32 18 12 2A B7 37 22 : 89
3808 AF 3A 2A D9 39 22 AD 3A : 2E
3810 3A BB 37 CD 55 33 3A 5B : 16
3818 1F CB 3F 67 2E 04 22 BC : A0
3820 37 C9 CD 55 33 18 03 CD : 3D
3828 E2 32 2A BC 37 25 22 BC : 34
3830 37 C9 2A D9 39 7E 23 FE : DB
3838 0D 20 FA ED 5B D9 39 18 : 99
3840 14 2A D9 39 7E 23 FE 0D : FC
3848 20 FA 7E 23 FE 0D 28 FA : E8
3850 2B ED 5B D9 39 7E 23 12 : 38
3858 13 A7 20 F9 1B ED 53 AB : D9
3860 3A CD F3 32 2A BC 37 25 : 6E
3868 22 BC 37 C9 2A B6 3A 7C : 74
3870 B5 CA C4 1F ED 5B AB 3A : 8F
3878 19 DA 1D 35 ED 72 D2 1D : 93
--------------------------------
SUM: F4 BB B0 73 E2 7E 4B CE 95BE

3880 35 2A AB 3A ED 4B D9 39 : 8E
3888 B7 ED 42 44 4D 03 2A B6 : 5A
3890 3A ED 5B AB 3A 19 22 AB : 4D
3898 3A EB ED B8 ED 4B B6 3A : F2
38A0 ED 5B D9 39 21 00 00 CD : 48
38A8 94 1F 23 12 13 0B 78 B1 : 2F
38B0 20 F5 CD F3 32 C9 2A E3 : DD
38B8 37 ED 5B B7 37 B7 ED 52 : 63
38C0 CA C4 1F 30 10 2A E3 37 : 31
38C8 ED 53 E3 37 22 B7 37 CD : 37
38D0 15 3A 22 D9 39 2A E3 37 : C7
38D8 CD 15 3A ED 5B D9 39 B7 : 2D
38E0 ED 52 22 B6 3A ED 5B 68 : 01
38E8 1F B7 ED 52 D2 1D 35 ED : 26
38F0 4B B6 3A ED 5B D9 39 21 : B6
38F8 00 00 1A 13 CD 9A 1F 23 : D6
--------------------------------
SUM: 28 70 1A 0B F8 9E 88 12 CCC0

3900 0B 78 B1 20 F5 2A E3 37 : 8D
3908 CD 15 3A ED 5B D9 39 7E : F4
3910 23 12 13 A7 20 F9 1B ED : 10
3918 53 AB 3A 2A B7 37 CD 15 : 32
3920 3A 22 BA 32 CD 4B 39 C9 : 62
3928 2A B7 37 22 AF 3A 3A 5B : B8
3930 1F CB 3F 16 00 5F B7 ED : 42
3938 52 30 03 21 01 00 22 AF : 78
3940 3A CD 15 3A 22 AD 3A CD : 2C
3948 F3 32 C9 2A B7 37 22 AF : D7
3950 3A 3A 5B 1F CB 3F 16 00 : 0E
3958 5F B7 ED 52 30 03 21 01 : AA
3960 00 22 AF 3A 22 B3 3A CD : E7
3968 15 3A 22 AD 3A 22 B1 3A : 65
3970 AF 32 54 33 3E 0C CD F4 : 73
3978 1F 2A B1 3A 7E A7 C8 CD : EE
--------------------------------
SUM: CC C6 67 92 90 C5 63 BC A2DC
```

```
3980 3D 33 3A 54 33 21 5B 1F : CC
3988 BE D0 2A B3 3A 23 22 B3 : 9D
3990 3A 2B CD DB 39 CD F1 1F : 23
3998 2A B1 3A ED 5B BA 32 1B : 64
39A0 7E E5 B7 ED 52 E1 20 08 : 62
39A8 E5 CD 18 20 22 BC 37 E1 : E0
39B0 CD F4 1F 23 FE 0D 20 E8 : 16
39B8 22 B1 3A 18 BC E5 D5 ED : 88
39C0 5B AB 3A 19 38 07 EB 2A : AD
39C8 6A 1F B7 ED 52 D1 E1 C9 : FA
39D0 ED 5B B9 37 1B 1A 13 B7 : 37
39D8 C9 00 00 F5 C5 D5 E5 06 : 43
39E0 04 11 03 3A AF 12 1B E5 : 13
39E8 CD 04 3A C6 30 12 1B 10 : 3E
39F0 F7 13 E1 23 11 FF 39 CD : 24
39F8 E5 1F E1 D1 C1 F1 C9 C5 : F6
```

```
---------------------------------
SUM: D9 A2 3C 3D 4A 35 E8 01 C7AD

3A00 0E 0A AF 06 C5 0E 0A AF : 59
3A08 06 10 29 8F B9 38 02 91 : 52
3A10 23 10 F7 C1 C9 D5 7D B4 : BA
3A18 20 01 23 54 5D 21 00 40 : 56
3A20 1B 7A B3 28 0B 7E B7 28 : D8
3A28 07 23 FE 0D 20 F7 18 F0 : 54
3A30 D1 C9 C5 21 00 00 3E 04 : C2
3A38 F5 1A 13 CD DB 32 38 13 : 47
3A40 29 4D 44 29 29 09 D6 30 : 1B
3A48 4F 06 00 09 F1 3D 20 E8 : 94
3A50 1A 13 F5 C1 C1 C9 CD A1 : DB
3A58 3A 3E 20 06 23 CD F4 1F : A1
3A60 10 FB CD A1 3A CD 80 3A : 3A
```

```
3A68 1A 13 2C FE 3A 20 F9 E5 : 8F
3A70 CD 1E 20 ED 5B 76 1F CD : B5
3A78 D3 1F E1 26 00 19 EB C9 : C6
---------------------------------
SUM: D5 9A CE 78 77 3B 08 F0 2DCD

3A80 F5 D5 1A 13 B7 28 17 FE : EB
3A88 5E CC F4 1F FE 20 30 09 : 94
3A90 F5 3E 5E CD F4 1F F1 C6 : 28
3A98 40 CD F4 1F 18 E4 D1 F1 : DE
3AA0 C9 3A 5B 1F 3D 2E 00 67 : 4F
3AA8 C3 1E 20 00 00 00 00 00 : 01
3AB0 00 00 00 00 00 00 00 00 : 00
3AB8 00 00 00                : 00
---------------------------------
SUM: 14 04 DB 3D FE 79 09 25 456C
```

リスト2

```
0000                ;
0000                ; SMALL Editor for S-OS "SWORD"
0000                ;
0000                ; Programed By Ishigami Tatsuya
0000                ;
0000                ;  8/15/92
0000                ;
0000
0000                        OFFSET  0A000H-3000H
3000                        ORG     3000H
3000                        ;
1FF4 P              PRINT   EQU     1FF4H
1FF1 P              PRNTS   EQU     1FF1H
1FEE P              LETNL   EQU     1FEEH
1FE8 P              MSG     EQU     1FE8H
1FE5 P              MSX     EQU     1FE5H
1FD3 P              GETL    EQU     1FD3H
1FD0 P              GETKY   EQU     1FD0H
1FC4 P              BELL    EQU     1FC4H
1FBE P              PRTHL   EQU     1FBEH
1FB5 P              AHEX    EQU     1FB5H
1FB2 P              HLHEX   EQU     1FB2H
1FAF P              WOPEN   EQU     1FAFH
1FAC P              WRD     EQU     1FACH
1FA6 P              RDD     EQU     1FA6H
1FA3 P              FILE    EQU     1FA3H
1F9D P              FPRNT   EQU     1F9DH
1F9A P              POKE    EQU     1F9AH
1F94 P              PEEK    EQU     1F94H
2006 P              DIR     EQU     2006H
2009 P              ROPEN   EQU     2009H
2021 P              FLGET   EQU     2021H
201E P              CSRSET  EQU     201EH
2018 P              CSRRD   EQU     2018H
2024 P              RDVSW   EQU     2024H
2033 P              ERROR   EQU     2033H
1F81 P              [HL]    EQU     1F81H
3000                        ;
1F76 P              KBFAD   EQU     1F76H
1F72 P              SIZE    EQU     1F72H
1F70 P              DTADR   EQU     1F70H
1F6E P              EXADR   EQU     1F6EH
1F6A P              MEMAX   EQU     1F6AH
1F68 P              WKSIZ   EQU     1F68H
1F5D P              DSK     EQU     1F5DH
1F5C P              WIDTH   EQU     1F5CH
1F5B P              MAXLN   EQU     1F5BH
3000
4000 P              #TXTST  EQU     4000H           ;Text Area Start
3000
3000 ED 73 B9 3A            LD      (SPBUF),SP
3004 ED 7B 6A 1F            LD      SP,(MEMAX)
3008 CD F0 30               CALL    GetCommandLine
300B
300B
300B AF                     XOR     A
300C 32 B8 3A               LD      (IsTxtChg),A
300F
300F 21 00 00               LD      HL,0
3012 22 B6 3A               LD      (LenCutBuf),HL
3015
3015 21 00 40               LD      HL,#TXTST
3018 22 AD 3A               LD      (#CURADR),HL
301B
301B 36 0D                  LD      (HL),0DH
301D 23                     INC     HL
301E 36 00                  LD      (HL),0
3020 22 AB 3A               LD      (#TXTEND),HL
3023                        ;
3023 21 01 00               LD      HL,1
3026 22 AF 3A               LD      (#CURLN),HL
3029 22 E3 37               LD      (MLNNO),HL
302C
302C 3A 66 35               LD      A,(FNAME)
302F A7                     AND     A
3030 28 0D                  JR      Z,COMMAND
3032
3032 3E 04                  LD      A,4
3034 11 66 35               LD      DE,FNAME
3037 CD A3 1F               CALL    FILE
303A CD E2 33               CALL    LOAD1
303D 18 03                  JR      CMND
303F                ;
303F                ; COMMAND LOOP
303F                ;
303F                COMMAND:
303F CD F3 32               CALL    CURPG
3042                CMND:
3042 CD A1 3A               CALL    CMNDLN
3045 3E 3E                  LD      A,'>'
3047 CD F4 1F               CALL    PRINT
304A CD 21 20               CALL    FLGET
304D E6 DF                  AND     0DFH            ;toupper
304F 21 71 30               LD      HL,COMTBL
3052 06 0D                  LD      B,13
3054 CD 5E 30               CALL    JPTBL
3057 38 E9                  JR      C,CMND
3059
3059 11 42 30               LD      DE,CMND         ;Return Address
305C D5                     PUSH    DE
305D E9                     JP      (HL)
305E
305E                JPTBL:
305E BE                     CP      (HL)
305F 20 09                  JR      NZ,JPTBL1
3061 23                     INC     HL
3062 F5                     PUSH    AF
3063 7E                     LD      A,(HL)
3064 23                     INC     HL
3065 66                     LD      H,(HL)
3066 6F                     LD      L,A
3067 F1                     POP     AF
3068 B7                     OR      A
3069 C9                     RET
306A                        ;
306A 23             JPTBL1: INC     HL
306B 23                     INC     HL
306C 23                     INC     HL
306D 10 EF                  DJNZ    JPTBL
306F 37                     SCF
3070 C9                     RET
3071                ;
3071                COMTBL:
3071 47                     DEFB    'G'
3072 BC 32                  DEFW    GOLINE
3074 56                     DEFB    'V'
3075 E2 32                  DEFW    NEXTPG
3077 5A                     DEFB    'Z'
3078 55 33                  DEFW    PREVPG
307A 45                     DEFB    'E'
307B CB 36                  DEFW    EDIT
307D 4C                     DEFB    'L'
307E F3 32                  DEFW    CURPG
3080 46                     DEFB    'F'
3081 B0 33                  DEFW    FILEIO
3083 53                     DEFB    'S'
3084 ED 31                  DEFW    SEARCH
3086 43                     DEFB    'C'
3087 48 31                  DEFW    REPLACE
3089 4D                     DEFB    'M'
308A 30 31                  DEFW    REPORT
308C 51                     DEFB    'Q'
308D 8F 30                  DEFW    QUIT
308F
308F                QUIT:
308F 3A B8 3A               LD      A,(IsTxtChg)
3092 A7                     AND     A
3093 C4 9F 30               CALL    NZ,MakeSure
3096 D8                     RET     C
3097 CD EE 1F               CALL    LETNL
309A ED 7B B9 3A            LD      SP,(SPBUF)
309E C9                     RET
309F
309F                MakeSure:
309F CD A1 3A               CALL    CMNDLN
30A2 11 D8 30               LD      DE,MSGBlank
30A5 CD E5 1F               CALL    MSX
30A8 CD A1 3A               CALL    CMNDLN
30AB 11 C4 30               LD      DE,MSGSure
30AE CD E5 1F               CALL    MSX
30B1 CD 21 20               CALL    FLGET
30B4 E6 DF                  AND     0DFH            ;toupper
30B6 FE 59                  CP      'Y'
30B8 C8                     RET     Z               ;with CY = 0
30B9 CD A1 3A               CALL    CMNDLN
30BC 11 D8 30               LD      DE,MSGBlank
30BF CD E5 1F               CALL    MSX
30C2 37                     SCF
30C3 C9                     RET
30C4
30C4                MSGSure:
30C4 41 72 65 20            DB      'Are You Sure ?(Y/N)',0
30C8 59 6F 75 20
30CC 53 75 72 65
30D0 20 3F 28 59
30D4 2F 4E 29 00
30D8                MSGBlank:
30D8 20 20 20 20            DB      20H,20H,20H,20H,20H,20H,20H,20H
30DC 20 20 20 20
30E0 20 20 20 20            DB      20H,20H,20H,20H,20H,20H,20H,20H
30E4 20 20 20 20
30E8 20 20 20 20            DB      20H,20H,20H,20H,20H,20H,20H,0
30EC 20 20 20 00
30F0
30F0                ;
30F0                ; Get Strings From Command Line
30F0                ;
30F0                GetCommandLine:
30F0 AF                     XOR     A
30F1 32 66 35               LD      (FNAME),A
30F4 3E 01                  LD      A,1
30F6 32 B5 3A               LD      (UPDATEFLAG),A
30F9
30F9 2A 76 1F               LD      HL,(KBFAD)
30FC 7E             Get1:   LD      A,(HL)
30FD 23                     INC     HL
30FE A7                     AND     A
30FF C8                     RET     Z
3100 FE 20                  CP      ' '
3102 20 F8                  JR      NZ,Get1
3104
3104 FE 2F                  CP      '/'
3106 28 1A                  JR      Z,Option
3108 FE 2D                  CP      '-'
310A 28 16                  JR      Z,Option
310C
310C 06 13                  LD      B,19
310E 11 66 35               LD      DE,FNAME
3111 7E             Get2:   LD      A,(HL)
3112 12                     LD      (DE),A
3113 A7                     AND     A
3114 28 08                  JR      Z,Get3
3116 FE 20                  CP      ' '
3118 28 04                  JR      Z,Get3
311A 13                     INC     DE
311B 23                     INC     HL
311C 18 F3                  JR      Get2
311E
311E AF             Get3:   XOR     A
311F 12                     LD      (DE),A
3120 18 DA                  JR      Get1
3122
```

▶SMは面白いです。えっ，もちろんセーラームーンに決まっているじゃないですか。
山中　政宣(18)三重県

```
3122 7E          Option: LD      A,(HL)
3123 E6 DF               AND     0DFH              ;toupper
3125 FE 4E               CP      'N'
3127 20 D3               JR      NZ,Get1
3129
3129 23                  INC     HL
312A AF                  XOR     A
312B 32 B5 3A            LD      (UPDATEFLAG),A
312E 18 CC               JR      Get1
3130
3130             ;
3130             ; MEMORY REPORT
3130             ;
3130 CD A1 3A    REPORT: CALL    CMNDLN
3133 CD F1 1F            CALL    PRNTS
3136 21 00 40            LD      HL,#TXTST
3139 CD BE 1F            CALL    PRTHL
313C 3E 2D               LD      A,'-'
313E CD F4 1F            CALL    PRINT
3141 2A AB 3A            LD      HL,(#TXTEND)
3144 CD BE 1F            CALL    PRTHL
3147 C9                  RET
3148             ;
3148             ; REPLACE FORWARD
3148             ;
3148             REPLACE:
3148 CD 41 32            CALL    SCH1
314B FE 1B               CP      1BH
314D C8                  RET     Z
314E 11 CF 31            LD      DE,REPMES
3151 CD 56 3A            CALL    GETCMND
3154 CD 87 32            CALL    BREAK?
3157 C8                  RET     Z
3158 1A                  LD      A,(DE)
3159 B7                  OR      A
315A C8                  RET     Z
315B 21 D6 31            LD      HL,REPSTR
315E CD 69 32            CALL    CNVTRNS
3161                     ;
3161 21 A5 32            LD      HL,SCHSTR
3164 CD 9E 33            CALL    STRLEN
3167 E5                  PUSH    HL
3168 21 D6 31            LD      HL,REPSTR
316B CD 9E 33            CALL    STRLEN
316E D1                  POP     DE
316F B7                  OR      A
3170 ED 52               SBC     HL,DE
3172 22 EB 31            LD      (XCHGSZ),HL
3175 CD 03 32    RPLC1:  CALL    ASK
3178 38 0B               JR      C,RPLC3
317A CD 21 20    RPLC2:  CALL    FLGET
317D FE 0D               CP      0DH
317F 28 08               JR      Z,XCHG
3181 FE 1B               CP      1BH
3183 20 F0               JR      NZ,RPLC1
3185 CD F3 32    RPLC3:  CALL    CURPG
3188 C9                  RET
3189                     ;
3189 3E 01       XCHG:   LD      A,1
318B 32 B8 3A            LD      (IsTxtChg),A
318E 2A EB 31            LD      HL,(XCHGSZ)
3191 7C                  LD      A,H
3192 B7                  OR      A
3193 F4 BD 39            CALL    P,SIZECHK
3196 DA 1D 35            JP      C,MEMERR
3199 2A EB 31            LD      HL,(XCHGSZ)
319C 7C                  LD      A,H
319D B5                  OR      L
319E 28 1E               JR      Z,XCHG2
31A0 E5                  PUSH    HL
31A1 7C                  LD      A,H
31A2 E6 80               AND     80H
31A4 20 0D               JR      NZ,XCHG1
31A6
31A6 2A BA 32            LD      HL,(SCHADR)       ;INC SIZE
31A9 2B                  DEC     HL
31AA 22 D9 39            LD      (LNADR),HL
31AD E1                  POP     HL
31AE CD 5C 37            CALL    INCSIZE
31B1 18 0B               JR      XCHG2
31B3
31B3 2A BA 32    XCHG1:  LD      HL,(SCHADR)       ;DEC SIZE
31B6 2B                  DEC     HL
31B7 22 D9 39            LD      (LNADR),HL
31BA E1                  POP     HL
31BB CD 77 37            CALL    DECSIZE
31BE
31BE 21 D6 31    XCHG2:  LD      HL,REPSTR
31C1 ED 5B BA 32         LD      DE,(SCHADR)
31C5 1B                  DEC     DE
31C6 7E          XCHG3:  LD      A,(HL)
31C7 B7                  OR      A
31C8 28 AB               JR      Z,RPLC1
31CA 23          XCHG4:  INC     HL
31CB 12                  LD      (DE),A
31CC 13                  INC     DE
31CD 18 F7               JR      XCHG3
31CF             ;
31CF 52 65 70 6C REPMES: DEFB    "Replce:"
31D3 63 65 3A
31D6 20 20 20 20 REPSTR: DEFB    "                    "
31DA 20 20 20 20
31DE 20 20
31E0 20 20 20 20         DEFB    "            ",0
31E4 20 20 20 20
31E8 20 20 00
31EB 00 00       XCHGSZ: DEFW    0
31ED             ;
31ED             ; SEARCH FORWARD
31ED             ;
31ED             SEARCH:
31ED CD 41 32            CALL    SCH1
31F0 FE 1B               CP      1BH
31F2 C8                  RET     Z
31F3 CD 03 32    SEARCH1:CALL    ASK
31F6 D8                  RET     C
31F7 CD 21 20    SEARCH2:CALL    FLGET
31FA FE 0D               CP      0DH
31FC 28 F5               JR      Z,SEARCH1
31FE FE 1B               CP      1BH
3200 20 F5               JR      NZ,SEARCH2
3202 C9                  RET
3203             ;
3203 2A AB 3A    ASK:    LD      HL,(#TXTEND)
3206 ED 5B BA 32         LD      DE,(SCHADR)
320A B7                  OR      A
320B ED 52               SBC     HL,DE
320D 7D                  LD      A,L
320E B4                  OR      H
320F 37                  SCF
3210 C8                  RET     Z                 ;if(#TXTEND == SCHADR)
                                                   ;RET with CY = 1
3211                     ;
3211 44                  LD      B,H
3212 4D                  LD      C,L
3213 EB                  EX      DE,HL
3214 11 A5 32            LD      DE,SCHSTR
3217 1A          SRCH2:  LD      A,(DE)
3218 ED B1               CPIR
321A 37                  SCF
321B C0                  RET     NZ                ;Not Found
321C                     ;
321C 22 BA 32            LD      (SCHADR),HL
321F 2B                  DEC     HL
3220 23          SRCH3:  INC     HL
3221 13                  INC     DE
3222 1A                  LD      A,(DE)
3223 B7                  OR      A
3224 28 05               JR      Z,SRCH4
3226 BE                  CP      (HL)
3227 28 F7               JR      Z,SRCH3
3229 18 D8               JR      ASK
322B 2A BA 32    SRCH4:  LD      HL,(SCHADR)
322E 2B                  DEC     HL
322F CD AD 36            CALL    ADTOLN
3232 22 B7 37            LD      (LNNO),HL
3235 CD 4B 39            CALL    EDMDLN2
3238 2A BC 37            LD      HL,(LOC)
323B CD 1E 20            CALL    CSRSET
323E B7                  OR      A                 ;CY = 0
323F C9                  RET
3240 C9                  RET
3241             ;
3241             SCH1:
3241 CD A1 3A            CALL    CMNDLN
3244 11 91 32            LD      DE,BCMES
3247 CD E5 1F            CALL    MSX
324A CD 21 20            CALL    FLGET
324D E6 DF               AND     0DFH              ; 1101.1111B
324F FE 42               CP      'B'
3251 2A AD 3A            LD      HL,(#CURADR)
3254 20 03               JR      NZ,SCH2
3256 21 00 40            LD      HL,#TXTST
3259 22 BA 32    SCH2:   LD      (SCHADR),HL
325C 11 9E 32            LD      DE,SCHMES
325F CD 56 3A            CALL    GETCMND
3262 CD 87 32            CALL    BREAK?
3265 C8                  RET     Z
3266                     ;
3266 21 A5 32            LD      HL,SCHSTR
3269             ;
3269             CNVTRNS:
3269             ;
3269 06 14               LD      B,20
326B 1A          CNVTRS1:LD      A,(DE)
326C 13                  INC     DE
326D 77                  LD      (HL),A
326E B7                  OR      A
326F C8                  RET     Z
3270 FE 5E               CP      '^'
3272 28 04               JR      Z,CNVTRS2
3274 23                  INC     HL
3275 10 F4               DJNZ    CNVTRS1
3277 C9                  RET
3278 1A          CNVTRS2:LD      A,(DE)
3279 13                  INC     DE
327A FE 5E               CP      '^'
327C 28 04               JR      Z,CNVTRS3
327E E6 DF               AND     0DFH              ; 1101.1111B
3280 D6 40               SUB     '@'
3282 77          CNVTRS3:LD      (HL),A
3283 23                  INC     HL
3284 10 E5               DJNZ    CNVTRS1
3286 C9                  RET
3287             ;
3287             BREAK?:
3287             ;
3287 D5                  PUSH    DE
3288 ED 5B 76 1F         LD      DE,(KBFAD)
328C 1A                  LD      A,(DE)
328D D1                  POP     DE
328E FE 1B               CP      1BH
3290 C9                  RET
3291             ;
3291 42 65 67 69 BCMES:  DEFB    "Begin, Here ",0
3295 6E 2C 20 48
3299 65 72 65 20
329D 00
329E 53 65 61 72 SCHMES: DEFB    "Search:"
32A2 63 68 3A
32A5 20 20 20 20 SCHSTR: DEFB    "                    "
32A9 20 20 20 20
32AD 20 20
32AF 20 20 20 20         DEFB    "            ", 0
32B3 20 20 20 20
32B7 20 20 00
32BA             ;
32BA 00 00       SCHADR: DEFW    0
32BC
32BC
32BC             ;
32BC             ; Go to Line
32BC             ;
32BC 11 D5 32    GOLINE: LD      DE,GOTOMES
32BF CD 56 3A            CALL    GETCMND
32C2 CD 87 32            CALL    BREAK?
32C5 C8                  RET     Z
32C6 CD 32 3A            CALL    GETLN
32C9 22 AF 3A            LD      (#CURLN),HL
32CC CD 15 3A            CALL    GETLNADR
32CF 22 AD 3A            LD      (#CURADR),HL
32D2 C3 F3 32            JP      CURPG
32D5             ;
32D5 47 6F 74 6F GOTOMES:DEFB    "Goto:", 0
32D9 3A 00
32DB
32DB             ;
32DB             ; Check if A is number
32DB             ;
32DB FE 30       ISDIGIT:CP      '0'
32DD D8                  RET     C
32DE FE 3A               CP      '9'+1
32E0 3F                  CCF
32E1 C9                  RET
32E2             ;
32E2             ; Print Next Page
32E2             ;
32E2 ED 5B B1 3A NEXTPG: LD      DE,(#NXTADR)
32E6 1A                  LD      A,(DE)
32E7 B7                  OR      A
32E8 C8                  RET     Z
32E9 ED 53 AD 3A         LD      (#CURADR),DE
32ED 2A B3 3A            LD      HL,(#NXTLN)
32F0 22 AF 3A            LD      (#CURLN),HL
32F3             ;       CALL    CURPG
32F3             ;       RET
32F3             ;
32F3             ; Print Page
32F3             ;
32F3 AF          CURPG:  XOR     A
32F4 32 54 33            LD      (LOC_Y),A
32F7 2A AF 3A            LD      HL,(#CURLN)
32FA 22 B3 3A            LD      (#NXTLN),HL
32FD 2A AD 3A            LD      HL,(#CURADR)
3300 22 B1 3A            LD      (#NXTADR),HL
3303             *
3303 3E 0C               LD      A,0CH             ; CLS
3305 CD F4 1F            CALL    PRINT
3308 2A B1 3A    PRPG1:  LD      HL,(#NXTADR)
330B 7E                  LD      A,(HL)
330C A7                  AND     A
330D C8                  RET     Z                 ;End of Text
330E
330E CD 3D 33            CALL    IncLOC_Y
3311 3A 54 33            LD      A,(LOC_Y)
3314 21 5B 1F            LD      HL,MAXLN
```

▶ヨンセンマンはベラボーマンやソニックブラストマンに似ている。

荒川　亮介(19)埼玉県

```
3317 BE                       CP      (HL)
3318 D0                       RET     NC
3319
3319 2A B3 3A                 LD      HL,(#NXTLN)
331C 23                       INC     HL
331D 22 B3 3A                 LD      (#NXTLN),HL
3320 2B                       DEC     HL
3321 CD DB 39                 CALL    PUTDEC
3324 CD F1 1F                 CALL    PRNTS
3327 ED 5B B1 3A              LD      DE,(#NXTADR)
332B CD E8 1F                 CALL    MSG
332E 1A           PRPG2:      LD      A,(DE)
332F 13                       INC     DE
3330 FE 0D                    CP      0DH
3332 20 FA                    JR      NZ,PRPG2
3334 ED 53 B1 3A              LD      (#NXTADR),DE
3338 CD EE 1F                 CALL    LETNL
333B 18 CB                    JR      PRPG1
333D
333D              ;
333D              ; LOC_Y += STRLEN(HL) / WIDTH
333D              ;
333D              IncLOC_Y:
333D E5                       PUSH    HL
333E CD 9E 33                 CALL    STRLEN
3341 3A 5C 1F                 LD      A,(WIDTH)
3344 67                       LD      H,A             ;H = WIDTH
3345 7D                       LD      A,L             ;A = STRLEN
3346 F5           IncLOC1:    PUSH    AF
3347 3A 54 33                 LD      A,(LOC_Y)
334A 3C                       INC     A
334B 32 54 33                 LD      (LOC_Y),A
334E F1                       POP     AF
334F 94                       SUB     H
3350 30 F4                    JR      NC,IncLOC1
3352 E1                       POP     HL
3353 C9                       RET
3354
3354              LOC_Y:  DS      1
3355
3355              ;
3355              ; Print Previous Page
3355              ;
3355              PREVPG:
3355 ED 5B AF 3A              LD      DE,(#CURLN)
3359 1B                       DEC     DE
335A 7A                       LD      A,D
335B B3                       OR      E
335C C8                       RET     Z               ;if #CURLN == 1
335D                          ;
335D AF                       XOR     A
335E 32 54 33                 LD      (LOC_Y),A
3361 2A AD 3A                 LD      HL,(#CURADR)
3364 2B                       DEC     HL
3365 22 AD 3A     PRVPG1:     LD      (#CURADR),HL
3368 CD 95 33                 CALL    BACKLINE
336B
336B E5                       PUSH    HL
336C CD 3D 33                 CALL    IncLOC_Y
336F 3A 54 33                 LD      A,(LOC_Y)
3372 21 5B 1F                 LD      HL,MAXLN
3375 BE                       CP      (HL)
3376 E1                       POP     HL
3377 38 EC                    JR      C,PRVPG1        ;while LOC_Y < MAXLN
3379
3379 2A AD 3A                 LD      HL,(#CURADR)
337C 11 00 40                 LD      DE,#TXTST
337F 37                       SCF
3380 ED 52                    SBC     HL,DE
3382 30 04                    JR      NC,PRVPG3       ;IF #CURADR >= #TXTST
3384 ED 53 AD 3A              LD      (#CURADR),DE
3388 2A AD 3A     PRVPG3:     LD      HL,(#CURADR)
338B CD AD 36                 CALL    ADTOLN
338E 22 AF 3A                 LD      (#CURLN),HL
3391 CD F3 32                 CALL    CURPG
3394 C9                       RET
3395              ;
3395              BACKLINE:
3395 2B                       DEC     HL
3396 2B           BACKL1:     DEC     HL
3397 7E                       LD      A,(HL)
3398 FE 0D                    CP      0DH
339A 20 FA                    JR      NZ,BACKL1
339C 23                       INC     HL
339D C9                       RET
339E              ;
339E              ; Count String Length
339E              ;
339E C5           STRLEN:     PUSH    BC
339F 01 FF FF                 LD      BC,-1
33A2 7E           STRL2:      LD      A,(HL)
33A3 23                       INC     HL
33A4 03                       INC     BC
33A5 FE 0D                    CP      0DH
33A7 28 03                    JR      Z,STRL1
33A9 A7                       AND     A
33AA 20 F6                    JR      NZ,STRL2
33AC 60           STRL1:      LD      H,B
33AD 69                       LD      L,C
33AE C1                       POP     BC
33AF C9                       RET
33B0
33B0              ;
33B0              ; File I/O
33B0              ;
33B0              FILEIO:
33B0 11 3D 35                 LD      DE,IOMES
33B3 CD E5 1F                 CALL    MSX
33B6 CD 21 20                 CALL    FLGET
33B9 E6 DF                    AND     0DFH            ; 1101,1111B
33BB FE 4C                    CP      'L'
33BD 28 17                    JR      Z,LOAD
33BF FE 53                    CP      'S'
33C1 CA 65 34                 JP      Z,SAVE
33C4 FE 43                    CP      'C'
33C6 CA 98 34                 JP      Z,CutSave
33C9 FE 47                    CP      'G'
33CB CA 17 34                 JP      Z,GetCut
33CE FE 44                    CP      'D'
33D0 CA 7A 35                 JP      Z,SELECT
33D3 C3 F3 32                 JP      CURPG
33D6
33D6              LOAD:
33D6 3A B8 3A                 LD      A,(IsTxtChg)
33D9 A7                       AND     A
33DA C4 9F 30                 CALL    NZ,MakeSure
33DD D8                       RET     C
33DE CD F9 34                 CALL    InputFileName
33E1 D8                       RET     C
33E2 CD 09 20     LOAD1:      CALL    ROPEN
33E5 DA 34 35                 JP      C,PRERMES
33E8 28 08                    JR      Z,LOAD3
33EA CD 9D 1F                 CALL    FPRNT
33ED CD EE 1F                 CALL    LETNL
33F0 18 E4                    JR      LOAD
33F2
33F2 21 01 00     LOAD3:      LD      HL,1
33F5 22 AF 3A                 LD      (#CURLN),HL
33F8 22 E3 37                 LD      (MLNNO),HL
33FB 21 00 40                 LD      HL,#TXTST
33FE 22 AD 3A                 LD      (#CURADR),HL
3401 22 70 1F                 LD      (DTADR),HL
3404 ED 5B 72 1F              LD      DE,(SIZE)
3408 19                       ADD     HL,DE
3409 2B                       DEC     HL
340A 22 AB 3A                 LD      (#TXTEND),HL
340D CD A6 1F                 CALL    RDD
3410 DA 34 35                 JP      C,PRERMES
3413 CD F3 32                 CALL    CURPG
3416 C9                       RET
3417
3417              GetCut:
3417 CD F9 34                 CALL    InputFileName
341A D8                       RET     C
341B CD 09 20                 CALL    ROPEN
341E DA 34 35                 JP      C,PRERMES
3421 2A AB 3A                 LD      HL,(#TXTEND)
3424 ED 5B 72 1F              LD      DE,(SIZE)
3428 19                       ADD     HL,DE
3429 DA 1D 35                 JP      C,MEMERR
342C EB                       EX      DE,HL
342D 2A 6A 1F                 LD      HL,(MEMAX)
3430 B7                       OR      A
3431 ED 52                    SBC     HL,DE
3433 DA 1D 35                 JP      C,MEMERR
3436
3436 2A AB 3A                 LD      HL,(#TXTEND)
3439 23                       INC     HL
343A 22 70 1F                 LD      (DTADR),HL
343D E5                       PUSH    HL
343E CD A6 1F                 CALL    RDD
3441 E1                       POP     HL
3442 DA 34 35                 JP      C,PRERMES
3445
3445 ED 4B 72 1F              LD      BC,(SIZE)
3449 0B                       DEC     BC
344A ED 43 B6 3A              LD      (LenCutBuf),BC
344E ED 5B AB 3A              LD      DE,(#TXTEND)
3452 13                       INC     DE
3453 21 00 00                 LD      HL,0
3456 1A           GetCut1:    LD      A,(DE)
3457 13                       INC     DE
3458 CD 9A 1F                 CALL    POKE
345B 23                       INC     HL
345C 0B                       DEC     BC
345D 78                       LD      A,B
345E B1                       OR      C
345F 20 F5                    JR      NZ,GetCut1
3461
3461 CD F3 32                 CALL    CURPG
3464 C9                       RET
3465              ;
3465              SAVE:
3465 CD F9 34                 CALL    InputFileName
3468 D8                       RET     C
3469
3469 2A AB 3A                 LD      HL,(#TXTEND)
346C 11 00 40                 LD      DE,#TXTST
346F B7                       OR      A
3470 ED 52                    SBC     HL,DE
3472 22 72 1F                 LD      (SIZE),HL
3475 21 00 00                 LD      HL,0
3478 22 70 1F                 LD      (DTADR),HL
347B 22 6E 1F                 LD      (EXADR),HL
347E CD AF 1F                 CALL    WOPEN
3481 DA 34 35                 JP      C,PRERMES
3484 21 00 40                 LD      HL,#TXTST
3487 22 70 1F                 LD      (DTADR),HL
348A CD AC 1F                 CALL    WRD
348D DA 34 35                 JP      C,PRERMES
3490 CD D7 35                 CALL    UPDATEFILE
3493 AF                       XOR     A
3494 32 B8 3A                 LD      (IsTxtChg),A
3497 C9                       RET
3498
3498              CutSave:
3498 2A B6 3A                 LD      HL,(LenCutBuf)
349B 7C                       LD      A,H
349C B5                       OR      L
349D CA C4 1F                 JP      Z,BELL
34A0
34A0 CD F9 34                 CALL    InputFileName
34A3 D8                       RET     C
34A4
34A4 ED 4B B6 3A              LD      BC,(LenCutBuf)
34A8 03                       INC     BC
34A9 11 00 40                 LD      DE,#TXTST
34AC 21 00 00                 LD      HL,0
34AF CD E7 34                 CALL    CHGTXT
34B2 1B                       DEC     DE
34B3 AF                       XOR     A
34B4 12                       LD      (DE),A
34B5
34B5 2A B6 3A                 LD      HL,(LenCutBuf)
34B8 23                       INC     HL
34B9 22 72 1F                 LD      (SIZE),HL
34BC 21 00 00                 LD      HL,0
34BF 22 70 1F                 LD      (DTADR),HL
34C2 22 6E 1F                 LD      (EXADR),HL
34C5 CD AF 1F                 CALL    WOPEN
34C8 38 6A                    JR      C,PRERMES
34CA 21 00 40                 LD      HL,#TXTST
34CD 22 70 1F                 LD      (DTADR),HL
34D0 CD AC 1F                 CALL    WRD
34D3 38 5F                    JR      C,PRERMES
34D5 CD D7 35                 CALL    UPDATEFILE
34D8
34D8 ED 4B B6 3A              LD      BC,(LenCutBuf)
34DC 03                       INC     BC
34DD 11 00 40                 LD      DE,#TXTST
34E0 21 00 00                 LD      HL,0
34E3 CD E7 34                 CALL    CHGTXT
34E6 C9                       RET
34E7
34E7              CHGTXT:
34E7 1A                       LD      A,(DE)
34E8 F5                       PUSH    AF
34E9 CD 94 1F                 CALL    PEEK
34EC 12                       LD      (DE),A
34ED F1                       POP     AF
34EE CD 9A 1F                 CALL    POKE
34F1 13                       INC     DE
34F2 23                       INC     HL
34F3 0B                       DEC     BC
34F4 78                       LD      A,B
34F5 B1                       OR      C
34F6 20 EF                    JR      NZ,CHGTXT
34F8 C9                       RET
34F9
34F9              ;
34F9              InputFileName:
34F9 11 61 35                 LD      DE,NOMES
34FC CD 56 3A                 CALL    GETCMND
34FF CD 87 32                 CALL    BREAK?
3502 37                       SCF                     ;C = 1
3503 C8                       RET     Z
3504
3504 01 13 00                 LD      BC,19
3507 11 05 00                 LD      DE,5
350A 2A 76 1F                 LD      HL,(KBFAD)
350D 19                       ADD     HL,DE           ;Skip 'FILE:'
350E 11 66 35                 LD      DE,FNAME
3511 ED B0                    LDIR
3513 3E 04                    LD      A,4
3515 11 66 35                 LD      DE,FNAME
3518 CD A3 1F                 CALL    FILE
```


▶またやってしまった。本当に自分が嫌になる。歯磨き粉のチューブを便器に落としてしまったのだ。それも今回は使い始めて数日しかたっていないものを。私は歯磨き粉の神様に嫌われているのだろうか。そのうち我が家にきた歯磨き粉は，みんな通らなくてはならない儀式になってしまうのだろうか。
坊農　誠(21)福井県

```
351B B7                       OR      A       ; C = 0
351C C9                       RET
351D                  ;
351D
351D CD C4 1F         MEMERR: CALL    BELL
3520 CD A1 3A                 CALL    CMNDLN
3523 11 29 35                 LD      DE,MERMES
3526 C3 E5 1F                 JP      MSX
3529                  ;
3529 20 4E 4F 20      MERMES: DEFB    " NO MEMORY", 0
352D 4D 45 4D 4F
3531 52 59 00
3534                  ;
3534                  PRERMES:
3534 CD 33 20                 CALL    ERROR
3537 CD 21 20                 CALL    FLGET
353A C3 F3 32                 JP      CURPG
353D                  ;
353D 4C 6F 61 64      IOMES:  DEFB    "Load, Save, Get-cut, Cut-save, Dir ", 0
3541 2C 20 53 61
3545 76 65 2C 20
3549 47 65 74 2D
354D 63 75 74 2C
3551 20 43 75 74
3555 2D 73 61 76
3559 65 2C 20 44
355D 69 72 20 00
3561 46 49 4C 45      NOMES:  DEFB    "FILE:"
3565 3A
3566 4E 4F 4E 41      FNAME:  DEFB    "NONAME
356A 4D 45 20 20
356E 20 20 20 20
3572 20 20 20 20
3576 20 20 20 00
357A
357A                  ; Select File Name From Dir
357A                  ;
357A                  SELECT:
357A 11 CF 35                 LD      DE,SELMES
357D CD E5 1F                 CALL    MSX
3580 CD 21 20                 CALL    FLGET
3583 FE 1B                    CP      1BH
3585 C8                       RET     Z
3586 FE 0D                    CP      0DH
3588 CC 24 20                 CALL    Z,RDVSW
358B 32 5D 1F                 LD      (DSK),A
358E                          ;
358E CD EE 1F                 CALL    LETNL
3591 CD 06 20                 CALL    DIR
3594 11 B2 35                 LD      DE,DIRMES
3597 CD E5 1F                 CALL    MSX
359A ED 5B 76 1F              LD      DE,(KBFAD)
359E CD D3 1F                 CALL    GETL
35A1 1A                       LD      A,(DE)
35A2 FE 1B                    CP      1BH
35A4 C8                       RET     Z
35A5 21 05 00                 LD      HL,5
35A8 19                       ADD     HL,DE
35A9 11 66 35                 LD      DE,FNAME
35AC 01 13 00                 LD      BC,19
35AF ED B0                    LDIR
35B1 C9                       RET
35B2                  ;
35B2 48 69 74 20      DIRMES: DEFB    "Hit CR on filename or BREAK",0DH,0
35B6 43 52 20 6F
35BA 6E 20 66 69
35BE 6C 65 6E 61
35C2 6D 65 20 6F
35C6 72 20 42 52
35CA 45 41 4B 0D
35CE 00
35CF                  ;
35CF 0D 44 72 69      SELMES: DEFB    0DH,"Drive:", 0
35D3 76 65 3A 00
35D7
35D7
35D7                  UPDATEFILE:
35D7 3A B5 3A                 LD      A,(UPDATEFLAG)
35DA A7                       AND     A
35DB C8                       RET     Z
35DC
35DC 3E 04                    LD      A,4
35DE 11 A2 36                 LD      DE,Update
35E1 CD A3 1F                 CALL    FILE
35E4 CD 09 20                 CALL    ROPEN
35E7 38 5A                    JR      C,UPDT1         ;File Not Found
35E9
35E9 2A 72 1F                 LD      HL,(SIZE)
35EC ED 5B AB 3A              LD      DE,(#TXTEND)
35F0 19                       ADD     HL,DE
35F1 B7                       OR      A
35F2 ED 72                    SBC     HL,SP
35F4 D2 1D 35                 JP      NC,MEMERR
35F7
35F7 13                       INC     DE
35F8 ED 53 70 1F              LD      (DTADR),DE
35FC CD A6 1F                 CALL    RDD
35FF DA 70 36                 JP      C,UpdateErr
3602
3602 ED 5B AB 3A              LD      DE,(#TXTEND)
3606 2A 72 1F                 LD      HL,(SIZE)
3609 23                       INC     HL
360A 19                       ADD     HL,DE
360B
360B 11 66 35                 LD      DE,FNAME
360E 1A               UPDT3:  LD      A,(DE)
360F 13                       INC     DE
3610 FE 20                    CP      ' '
3612 28 FA                    JR      Z,UPDT3         ;SKIP SPACE
3614 77                       LD      (HL),A
3615 23                       INC     HL
3616 A7                       AND     A
3617 20 F5                    JR      NZ,UPDT3
3619
3619 77                       LD      (HL),A
361A 2B                       DEC     HL
361B 36 0D                    LD      (HL),0DH
361D 23                       INC     HL
361E
361E ED 5B AB 3A              LD      DE,(#TXTEND)
3622 B7                       OR      A
3623 ED 52                    SBC     HL,DE
3625 23                       INC     HL
3626 22 72 1F                 LD      (SIZE),HL
3629 21 00 00                 LD      HL,0
362C 22 70 1F                 LD      (DTADR),HL
362F 22 6E 1F                 LD      (EXADR),HL
3632 CD AF 1F                 CALL    WOPEN
3635 38 39                    JR      C,UpdateErr
3637 2A AB 3A                 LD      HL,(#TXTEND)
363A 22 70 1F                 LD      (DTADR),HL
363D CD AC 1F                 CALL    WRD
3640 38 2E                    JR      C,UpdateErr
3642 C9                       RET
3643
3643                  ;
3643                  ; Create 'UPDATE.$$$'
3643                  ;
3643 21 66 35         UPDT1:  LD      HL,FNAME
3646 CD 77 36                 CALL    INSTCR          ;Insert CR
3649 CD 9E 33                 CALL    STRLEN
364C 23                       INC     HL
364D 22 72 1F                 LD      (SIZE),HL
3650 21 00 00                 LD      HL,0
3653 22 70 1F                 LD      (DTADR),HL
3656 22 6E 1F                 LD      (EXADR),HL
3659 CD AF 1F                 CALL    WOPEN
365C 38 12                    JR      C,UpdateErr
365E 21 66 35                 LD      HL,FNAME
3661 22 70 1F                 LD      (DTADR),HL
3664 CD AC 1F                 CALL    WRD
3667 38 07                    JR      C,UpdateErr
3669 21 6B 35                 LD      HL,FNAME+5
366C CD 82 36                 CALL    DELCR           ;Delete CR
366F C9                       RET
3670
3670                  UpdateErr:
3670 11 8E 36                 LD      DE,UpdateErrMsg
3673 CD E5 1F                 CALL    MSX
3676 C9                       RET
3677
3677                  INSTCR:
3677 E5                       PUSH    HL
3678 7E               INS1:   LD      A,(HL)
3679 23                       INC     HL
367A A7                       AND     A
367B 20 FB                    JR      NZ,INS1
367D 2B                       DEC     HL
367E 36 0D                    LD      (HL),0DH
3680 E1                       POP     HL
3681 C9                       RET
3682
3682 E5               DELCR:  PUSH    HL
3683 7E               DEL1:   LD      A,(HL)
3684 23                       INC     HL
3685 FE 0D                    CP      0DH
3687 20 FA                    JR      NZ,DEL1
3689 2B                       DEC     HL
368A 36 00                    LD      (HL),0
368C E1                       POP     HL
368D C9                       RET
368E
368E                  UpdateErrMsg:
368E 46 49 4C 45              DB      'FILE ACCESS ERROR ON'
3692 20 41 43 43
3696 45 53 53 20
369A 45 52 52 4F
369E 52 20 4F 4E
36A2 55 50 44 41      Update: DB      'UPDATE.$$$',0
36A6 54 45 2E 24
36AA 24 24 00
36AD                  ;
36AD                  ; Convert Address(HL) to Line-NO
36AD                  ;
36AD C5               ADTOLN: PUSH    BC
36AE D5                       PUSH    DE
36AF 01 00 00                 LD      BC,0
36B2 11 00 40                 LD      DE,#TXTST
36B5                          ;
36B5 1A               adtoln1:LD      A,(DE)          ;while(HL > DE)
36B6 A7                       AND     A
36B7 28 0D                    JR      Z,adtoln4
36B9 13                       INC     DE
36BA FE 0D                    CP      0DH
36BC 20 F7                    JR      NZ,adtoln1
36BE
36BE 03                       INC     BC
36BF
36BF E5                       PUSH    HL
36C0 B7                       OR      A
36C1 ED 52                    SBC     HL,DE
36C3 E1                       POP     HL
36C4 30 EF                    JR      NC,adtoln1
36C6                          ;
36C6 60               adtoln4:LD      H,B
36C7 69                       LD      L,C
36C8 D1                       POP     DE
36C9 C1                       POP     BC
36CA C9                       RET
36CB
36CB                  ; Edit
36CB                  ;  Continue edittig until 'BREAK'ed
36CB                  ;
36CB                  EDIT:
36CB 21 04 00                 LD      HL,4
36CE 22 BC 37                 LD      (LOC),HL
36D1                  ;
36D1                  EDIT1:  ; EDIT entry
36D1 2A BC 37                 LD      HL,(LOC)
36D4 CD 1E 20                 CALL    CSRSET
36D7 ED 5B 76 1F              LD      DE,(KBFAD)
36DB CD D3 1F                 CALL    GETL
36DE CD 18 20                 CALL    CSRRD
36E1 2E 04                    LD      L,4
36E3 22 BC 37                 LD      (LOC),HL
36E6 1A                       LD      A,(DE)
36E7 FE 1B                    CP      1BH             ; 'BREAK'ed ?
36E9 CA A1 3A                 JP      Z,CMNDLN
36EC CD DB 32                 CALL    ISDIGIT
36EF 38 E0                    JR      C,EDIT1
36F1
36F1 3E 01                    LD      A,1
36F3 32 B8 3A                 LD      (IsTxtChg),A
36F6
36F6 CD 32 3A                 CALL    GETLN
36F9 22 B7 37                 LD      (LNNO),HL
36FC 32 BB 37                 LD      (CTRLCHAR),A    ; save cntrol char
36FF ED 53 B9 37              LD      (CTRLCOL),DE
3703 CD 15 3A                 CALL    GETLNADR        ; HL=ADR
3706 22 D9 39                 LD      (LNADR),HL      ; save lnadrs
3709
3709 21 D1 36                 LD      HL,EDIT1
370C E5                       PUSH    HL
370D
370D 3A BB 37                 LD      A,(CTRLCHAR)
3710 06 0A                    LD      B,10
3712 21 BE 37                 LD      HL,EDTBL
3715 CD 5E 30                 CALL    JPTBL
3718 38 01                    JR      C,EDIT3
371A E9                       JP      (HL)
371B                          ;
371B 2A D9 39         EDIT3:  LD      HL,(LNADR)
371E CD 9E 33                 CALL    STRLEN
3721 E5                       PUSH    HL
3722 2A B9 37                 LD      HL,(CTRLCOL)
3725 CD 9E 33                 CALL    STRLEN          ;HL = New Line Size
3728 3A BB 37                 LD      A,(CTRLCHAR)
372B FE 2B                    CP      '+'             ; Check CTRLCHR
372D 20 04                    JR      NZ,EDIT5
372F 11 0A 00                 LD      DE,10
3732 19                       ADD     HL,DE
3733 D1               EDIT5:  POP     DE              ;DE : Old Line Size
3734 B7                       OR      A
3735 ED 52                    SBC     HL,DE           ; HL=DIFFERENCE
3737 7D                       LD      A,L
3738 B4                       OR      H
3739 28 54                    JR      Z,TRNS
373B EB                       EX      DE,HL
373C 2A B1 3A                 LD      HL,(#NXTADR)
373F 19                       ADD     HL,DE
3740 22 B1 3A                 LD      (#NXTADR),HL    ;#NXTADR += Difference
3743 EB                       EX      DE,HL
3744 7C                       LD      A,H
3745 E6 80                    AND     80H             ; 1000.0000B
3747 C2 55 37                 JP      NZ,EDIT6
374A CD BD 39                 CALL    SIZECHK
```

▶ 9月18日，私はOh!Xを買いに本屋へ寄った。Oh!Xを手に取りレジへ持っていく。店員さんがレジを叩き「600円です」といいながら紙袋へ入れてくれる。財布を出そうと鞄の中へ手を入れる。すると財布に金が入っていないことに気づきハッとする。半年に一度はこれをやってしまうんだなあ。　　音羽　進(18)宮城県

```
374D D8                          RET     C
374E CD 5C 37                    CALL    INCSIZE
3751 CD 8F 37                    CALL    TRNS
3754 C9                          RET
3755
3755 CD 77 37       EDIT6:       CALL    DECSIZE
3758 CD 8F 37                    CALL    TRNS
375B C9                          RET
375C
375C ED 5B AB 3A    INCSIZE:LD           DE,(#TXTEND)
3760 D5                          PUSH    DE
3761 19                          ADD     HL,DE
3762 22 AB 3A                    LD      (#TXTEND),HL     ;#TXTEND += Difference
3765 E1                          POP     HL
3766 ED 4B D9 39                 LD      BC,(LNADR)
376A B7                          OR      A
376B ED 42                       SBC     HL,BC
376D 23                          INC     HL
376E 44                          LD      B,H
376F 4D                          LD      C,L              ;BC=TrnsSize
3770 2A AB 3A                    LD      HL,(#TXTEND)     ;DE=Old #TXTEND
3773 EB                          EX      DE,HL
3774 ED B8                       LDDR                     ;HL=New #TXTEND
3776 C9                          RET
3777                ;
3777 EB             DECSIZE:EX           DE,HL
3778 2A D9 39                    LD      HL,(LNADR)
377B B7                          OR      A
377C ED 52                       SBC     HL,DE            ;HL=LNADR + abs(Diffrenc
e)
377E ED 5B D9 39                 LD      DE,(LNADR)
3782
3782 7E             DECS1:       LD      A,(HL)
3783 23                          INC     HL
3784 12                          LD      (DE),A
3785 13                          INC     DE
3786 A7                          AND     A
3787 20 F9                       JR      NZ,DECS1
3789 1B                          DEC     DE
378A ED 53 AB 3A                 LD      (#TXTEND),DE
378E C9                          RET
378F
378F                ;
378F                ;
378F 2A D9 39       TRNS:        LD      HL,(LNADR)
3792 CD D0 39                    CALL    EMPLN
3795 28 09                       JR      Z,TRNS2
3797 1A             TRNS1:       LD      A,(DE)
3798 B7                          OR      A
3799 28 05                       JR      Z,TRNS2
379B 77                          LD      (HL),A
379C 23                          INC     HL
379D 13                          INC     DE
379E 18 F7                       JR      TRNS1
37A0 36 0D          TRNS2:       LD      (HL),0DH
37A2 23                          INC     HL
37A3                             ;
37A3 3A BB 37                    LD      A,(CTRLCHAR)
37A6 FE 2E                       CP      '.'
37A8 28 32                       JR      Z,EDMARK
37AA FE 2B                       CP      '+'
37AC C0                          RET     NZ
37AD 06 0A                       LD      B,10
37AF 36 0D          TRNS3:       LD      (HL),0DH
37B1 23                          INC     HL
37B2 10 FB                       DJNZ    TRNS3
37B4 C3 F3 32                    JP      CURPG
37B7                ;
37B7 00 00          LNNO:        DEFW    0
37B9 00 00          CTRLCOL:DEFW         0
37BB                CTRLCHAR:DS          1
37BC 00 00          LOC:         DEFW    0
37BE
37BE                EDTBL:
37BE 2D                          DEFB    '-'
37BF 32 38                       DEFW    DELETE
37C1 2F                          DEFB    '/'
37C2 41 38                       DEFW    DELETES
37C4 3E                          DEFB    '>'
37C5 6C 38                       DEFW    EDYANK
37C7 3C                          DEFB    '<'
37C8 B6 38                       DEFW    EDDR
37CA 3D                          DEFB    '='
37CB 28 39                       DEFW    EDMDLN
37CD 28                          DEFB    '('
37CE 22 38                       DEFW    SCRLD
37D0 29                          DEFB    ')'
37D1 27 38                       DEFW    SCRLU
37D3 2C                          DEFB    ','
37D4 E5 37                       DEFW    EXCHNGE
37D6 5B                          DEFB    '['
37D7 F3 37                       DEFW    MCLTOP
37D9 5D                          DEFB    ']'
37DA 04 38                       DEFW    MCLBTM
37DC                ;
37DC                ; MARK LINE
37DC                ;
37DC 2A B7 37       EDMARK:      LD      HL,(LNNO)
37DF 22 E3 37                    LD      (MLNNO),HL
37E2 C9                          RET
37E3                ;
37E3 00 00          MLNNO:       DEFW    0
37E5
37E5                ; Exchange Line and Mark
37E5                ;
37E5 2A E3 37       EXCHNGE:LD           HL,(MLNNO)
37E8 ED 5B B7 37                 LD      DE,(LNNO)
37EC ED 53 E3 37                 LD      (MLNNO),DE
37F0 C3 2B 39                    JP      EDMDLN1
37F3                ;
37F3                ; Move Current Line to Top
37F3                ;
37F3 2A B7 37       MCLTOP:      LD      HL,(LNNO)
37F6 22 AF 3A                    LD      (#CURLN),HL
37F9 2A D9 39                    LD      HL,(LNADR)
37FC 22 AD 3A                    LD      (#CURADR),HL
37FF CD F3 32                    CALL    CURPG
3802 18 12                       JR      MCLBTM1
3804                ;
3804                ; Move Current Line to Bottom
3804                ;
3804 2A B7 37       MCLBTM:      LD      HL,(LNNO)
3807 22 AF 3A                    LD      (#CURLN),HL
380A 2A D9 39                    LD      HL,(LNADR)
380D 22 AD 3A                    LD      (#CURADR),HL
3810 3A BB 37                    LD      A,(CTRLCHAR)
3813 CD 55 33                    CALL    PREVPG
3816 3A 5B 1F       MCLBTM1:LD           A,(MAXLN)
3819 CB 3F                       SRL     A
381B 67                          LD      H,A
381C 2E 04                       LD      L,4
381E 22 BC 37                    LD      (LOC),HL
3821 C9                          RET
3822                ;
3822                ; Previous & Next Page
3822                ;
3822                SCRLD:
3822 CD 55 33                    CALL    PREVPG
3825 18 03                       JR      SCR1
3827 CD E2 32       SCRLU:       CALL    NEXTPG
382A 2A BC 37       SCR1:        LD      HL,(LOC)
382D 25                          DEC     H
382E 22 BC 37                    LD      (LOC),HL
3831 C9                          RET
3832                ;
3832                ; Delete One line
3832                ;
3832 2A D9 39       DELETE:      LD      HL,(LNADR)
3835 7E             DLTM1:       LD      A,(HL)           ;HL = Endof(LNADR)
3836 23                          INC     HL
3837 FE 0D                       CP      0DH
3839 20 FA                       JR      NZ,DLTM1
383B
383B ED 5B D9 39                 LD      DE,(LNADR)
383F 18 14                       JR      DLTM2
3841
3841                ;
3841                ; Delete some lines
3841                ;
3841 2A D9 39       DELETES:LD           HL,(LNADR)
3844 7E             DELETE1:LD           A,(HL)
3845 23                          INC     HL
3846 FE 0D                       CP      0DH
3848 20 FA                       JR      NZ,DELETE1
384A                             ;
384A 7E             DELETE2:LD           A,(HL)
384B 23                          INC     HL
384C FE 0D                       CP      0DH
384E 28 FA                       JR      Z,DELETE2
3850 2B                          DEC     HL
3851 ED 5B D9 39                 LD      DE,(LNADR)
3855
3855 7E             DLTM2:       LD      A,(HL)
3856 23                          INC     HL
3857 12                          LD      (DE),A
3858 13                          INC     DE
3859 A7                          AND     A
385A 20 F9                       JR      NZ,DLTM2
385C
385C 1B                          DEC     DE
385D ED 53 AB 3A                 LD      (#TXTEND),DE
3861 CD F3 32                    CALL    CURPG
3864 2A BC 37                    LD      HL,(LOC)
3867 25                          DEC     H
3868 22 BC 37                    LD      (LOC),HL
386B C9                          RET
386C                ;
386C                ; Yank Delete Buffer
386C                ;
386C 2A B6 3A       EDYANK:      LD      HL,(LenCutBuf)
386F 7C                          LD      A,H
3870 B5                          OR      L
3871 CA C4 1F                    JP      Z,BELL
3874 ED 5B AB 3A                 LD      DE,(#TXTEND)
3878 19                          ADD     HL,DE
3879 DA 1D 35                    JP      C,MEMERR
387C                ;            OR      A
387C ED 72                       SBC     HL,SP
387E D2 1D 35                    JP      NC,MEMERR
3881                             ;
3881 2A AB 3A                    LD      HL,(#TXTEND)
3884 ED 4B D9 39                 LD      BC,(LNADR)
3888 B7                          OR      A
3889 ED 42                       SBC     HL,BC
388B 44                          LD      B,H
388C 4D                          LD      C,L              ;BC = #TXTEND - LNADR
388D 03                          INC     BC
388E 2A B6 3A                    LD      HL,(LenCutBuf)
3891 ED 5B AB 3A                 LD      DE,(#TXTEND)     ;DE = #TXTEND
3895 19                          ADD     HL,DE            ;HL = LenCutBuf + #TXTEN
D
3896 22 AB 3A                    LD      (#TXTEND),HL
3899 EB                          EX      DE,HL
389A                             ;
389A ED B8                       LDDR
389C                             ;
389C ED 4B B6 3A                 LD      BC,(LenCutBuf)
38A0 ED 5B D9 39                 LD      DE,(LNADR)
38A4 21 00 00                    LD      HL,0
38A7 CD 94 1F       EDYANK4:CALL         PEEK
38AA 23                          INC     HL
38AB 12                          LD      (DE),A
38AC 13                          INC     DE
38AD 0B                          DEC     BC
38AE 78                          LD      A,B
38AF B1                          OR      C
38B0 20 F5                       JR      NZ,EDYANK4
38B2 CD F3 32                    CALL    CURPG
38B5 C9                          RET
38B6                ;
38B6                ; Delete Region
38B6                ;
38B6 2A E3 37       EDDR:        LD      HL,(MLNNO)
38B9 ED 5B B7 37                 LD      DE,(LNNO)
38BD B7                          OR      A
38BE ED 52                       SBC     HL,DE
38C0 CA C4 1F                    JP      Z,BELL           ;There is no region
38C3 30 10                       JR      NC,EDDR1
38C5                             ;
38C5 2A E3 37                    LD      HL,(MLNNO)       ; case of LN > MARK
38C8 ED 53 E3 37                 LD      (MLNNO),DE       ;  ex LN, MARK
38CC 22 B7 37                    LD      (LNNO),HL
38CF CD 15 3A                    CALL    GETLNADR
38D2 22 D9 39                    LD      (LNADR),HL
38D5                             ;
38D5 2A E3 37       EDDR1:       LD      HL,(MLNNO)       ;Sure  LN < MARK
38D8 CD 15 3A                    CALL    GETLNADR
38DB ED 5B D9 39                 LD      DE,(LNADR)
38DF B7                          OR      A
38E0 ED 52                       SBC     HL,DE
38E2 22 B6 3A                    LD      (LenCutBuf),HL
38E5 ED 5B 68 1F                 LD      DE,(WKSIZ)
38E9 B7                          OR      A
38EA ED 52                       SBC     HL,DE
38EC D2 1D 35                    JP      NC,MEMERR
38EF
38EF ED 4B B6 3A                 LD      BC,(LenCutBuf)
38F3 ED 5B D9 39                 LD      DE,(LNADR)
38F7 21 00 00                    LD      HL,0
38FA 1A             EDDR2:       LD      A,(DE)
38FB 13                          INC     DE
38FC CD 9A 1F                    CALL    POKE
38FF 23                          INC     HL
3900 0B                          DEC     BC
3901 78                          LD      A,B
3902 B1                          OR      C
3903 20 F5                       JR      NZ,EDDR2
3905
3905 2A E3 37                    LD      HL,(MLNNO)
3908 CD 15 3A                    CALL    GETLNADR
390B ED 5B D9 39                 LD      DE,(LNADR)
390F 7E             EDDR3:       LD      A,(HL)
3910 23                          INC     HL
3911 12                          LD      (DE),A
3912 13                          INC     DE
3913 A7                          AND     A
3914 20 F9                       JR      NZ,EDDR3
3916
3916 1B                          DEC     DE
3917 ED 53 AB 3A                 LD      (#TXTEND),DE
391B
391B 2A B7 37                    LD      HL,(LNNO)
391E CD 15 3A                    CALL    GETLNADR
3921 22 BA 32                    LD      (SCHADR),HL
3924
3924 CD 4B 39                    CALL    EDMDLN2
```


▶ YMOの「テクノバイブル」を買いました。なんかこういっては失礼ですが，昔のナムコPSGで音が出ているようでとってもいいんですよね。10,000円は高くないですよ。リミックスはへぼかったけど。私もこれにならって人を感動させるような曲を作りたい。

戸辺　靖(18)栃木県

```
3927 C9                 RET
3928            ;
3928            ; Print line at middle
3928            ;
3928 2A B7 37   EDMDLN: LD      HL,(LNNO)
392B 22 AF 3A   EDMDLN1:LD      (#CURLN),HL
392E 3A 5B 1F           LD      A,(MAXLN)
3931 CB 3F              SRL     A
3933 16 00              LD      D,0
3935 5F                 LD      E,A
3936 B7                 OR      A
3937 ED 52              SBC     HL,DE           ;HL = CURLN - MAXLN / 2
3939 30 03              JR      NC,EDMD1
393B 21 01 00           LD      HL,1
393E 22 AF 3A   EDMD1:  LD      (#CURLN),HL
3941 CD 15 3A           CALL    GETLNADR
3944 22 AD 3A           LD      (#CURADR),HL
3947 CD F3 32           CALL    CURPG
394A C9                 RET
394B            ;
394B            ; Print line at middle with CURSSOR Set
394B            ;
394B 2A B7 37   EDMDLN2:LD      HL,(LNNO)
394E 22 AF 3A           LD      (#CURLN),HL
3951 3A 5B 1F           LD      A,(MAXLN)
3954 CB 3F              SRL     A
3956 16 00              LD      D,0
3958 5F                 LD      E,A
3959 B7                 OR      A
395A ED 52              SBC     HL,DE           ;HL = CURLN - MAXLN / 2
395C 30 03              JR      NC,EDMD2
395E 21 01 00           LD      HL,1
3961 22 AF 3A   EDMD2:  LD      (#CURLN),HL
3964 22 B3 3A           LD      (#NXTLN),HL
3967 CD 15 3A           CALL    GETLNADR
396A 22 AD 3A           LD      (#CURADR),HL
396D 22 B1 3A           LD      (#NXTADR),HL
3970 AF                 XOR     A
3971 32 54 33           LD      (LOC_Y),A
3974
3974 3E 0C              LD      A,0CH           ; CLS
3976 CD F4 1F           CALL    PRINT
3979 2A B1 3A   EDMD3:  LD      HL,(#NXTADR)
397C 7E                 LD      A,(HL)
397D A7                 AND     A
397E C8                 RET     Z               ;End of Text
397F
397F CD 3D 33           CALL    IncLOC_Y
3982 3A 54 33           LD      A,(LOC_Y)
3985 21 5B 1F           LD      HL,MAXLN
3988 BE                 CP      (HL)
3989 D0                 RET     NC
398A
398A 2A B3 3A           LD      HL,(#NXTLN)
398D 23                 INC     HL
398E 22 B3 3A           LD      (#NXTLN),HL
3991 2B                 DEC     HL
3992 CD DB 39           CALL    PUTDEC
3995 CD F1 1F           CALL    PRNTS
3998 2A B1 3A           LD      HL,(#NXTADR)
399B ED 5B BA 32        LD      DE,(SCHADR)
399F 1B                 DEC     DE
39A0 7E         EDMD4:  LD      A,(HL)
39A1 E5                 PUSH    HL              ;if(HL == (SCHADR)) LOC
= CSR
39A2 B7                 OR      A
39A3 ED 52              SBC     HL,DE
39A5 E1                 POP     HL
39A6 20 08              JR      NZ,EDMD5
39A8 E5                 PUSH    HL
39A9 CD 18 20           CALL    CSRRD
39AC 22 BC 37           LD      (LOC),HL
39AF E1                 POP     HL
39B0 CD F4 1F   EDMD5:  CALL    PRINT
39B3 23                 INC     HL
39B4 FE 0D              CP      0DH
39B6 20 E8              JR      NZ,EDMD4
39B8 22 B1 3A           LD      (#NXTADR),HL
39BB 18 BC              JR      EDMD3
39BD            ;
39BD            ; Size Check
39BD            ;   in:HL<-SIZE
39BD            ;
39BD E5         SIZECHK:PUSH    HL
39BE D5                 PUSH    DE
39BF ED 5B AB 3A        LD      DE,(#TXTEND)
39C3 19                 ADD     HL,DE
39C4 38 07              JR      C,SZCHK1
39C6 EB                 EX      DE,HL
39C7 2A 6A 1F           LD      HL,(MEMAX)
39CA B7                 OR      A
39CB ED 52              SBC     HL,DE
39CD D1         SZCHK1: POP     DE
39CE E1                 POP     HL
39CF C9                 RET
39D0            ;
39D0            ; Check line is empty
39D0            ;
39D0 ED 5B B9 37 EMPLN: LD      DE,(CTRLCOL)
39D4 1B                 DEC     DE
39D5 1A                 LD      A,(DE)
39D6 13                 INC     DE
39D7 B7                 OR      A
39D8 C9                 RET
39D9            ;
39D9 00 00      LNADR:  DEFW    0
39DB
39DB            ;
39DB            ; PUT OUT HL IN DECIMAL NUMBER
39DB            ;
39DB F5         PUTDEC: PUSH    AF
39DC C5                 PUSH    BC
39DD D5                 PUSH    DE
39DE E5                 PUSH    HL
39DF                    ;
39DF 06 04              LD      B,4             ; loop counter
39E1 11 03 3A           LD      DE,PLNOB+4
39E4 AF                 XOR     A
39E5 12                 LD      (DE),A
39E6 1B                 DEC     DE
39E7 E5                 PUSH    HL
39E8 CD 04 3A   PLNO1:  CALL    DIV10
39EB C6 30              ADD     A,'0'
39ED 12                 LD      (DE),A
39EE 1B                 DEC     DE
39EF 10 F7              DJNZ    PLNO1
39F1 13                 INC     DE
39F2 E1                 POP     HL
39F3 23                 INC     HL
39F4                    ;
39F4 11 FF 39           LD      DE,PLNOB
39F7 CD E5 1F           CALL    MSX
39FA E1                 POP     HL
39FB D1                 POP     DE
39FC C1                 POP     BC
39FD F1                 POP     AF
39FE C9                 RET
39FF
39FF            PLNOB:  DS      5
3A04
3A04            ;
3A04            ; DIVIDED BY 10
3A04            ;
3A04 C5         DIV10:  PUSH    BC
3A05 0E 0A              LD      C,10
3A07 AF                 XOR     A               ; A=0
3A08 06 10              LD      B,16            ; loop counter
3A0A 29         DIV101: ADD     HL,HL           ; shift left HL
3A0B 8F                 ADC     A,A
3A0C B9                 CP      C
3A0D 38 02              JR      C,DIV102
3A0F 91                 SUB     C               ; A-10 or C
3A10 23                 INC     HL              ; SetAnsBit
3A11 10 F7      DIV102: DJNZ    DIV101
3A13 C1                 POP     BC
3A14 C9                 RET
3A15            ;
3A15            ; Get ADRS of HL Line
3A15            ;
3A15            GETLNADR:
3A15 D5                 PUSH    DE
3A16 7D                 LD      A,L
3A17 B4                 OR      H
3A18 20 01              JR      NZ,GETLAD1
3A1A 23                 INC     HL              ;Sure HL >= 1
3A1B 54         GETLAD1:LD      D,H
3A1C 5D                 LD      E,L             ;DE = Target Ln
3A1D
3A1D 21 00 40           LD      HL,#TXTST
3A20 1B         GETLAD2:DEC     DE
3A21 7A                 LD      A,D
3A22 B3                 OR      E
3A23 28 0B              JR      Z,GETLAD4
3A25 7E         GETLAD3:LD      A,(HL)
3A26 B7                 OR      A
3A27 28 07              JR      Z,GETLAD4
3A29 23                 INC     HL
3A2A FE 0D              CP      0DH
3A2C 20 F7              JR      NZ,GETLAD3
3A2E 18 F0              JR      GETLAD2
3A30 D1         GETLAD4:POP     DE
3A31 C9                 RET
3A32
3A32            ;
3A32            ; Get Line No.
3A32            ;
3A32 C5         GETLN:  PUSH    BC
3A33 21 00 00           LD      HL,0            ; ANS
3A36 3E 04              LD      A,4             ; LOOP COUNTER
3A38 F5         GETLN1: PUSH    AF
3A39 1A                 LD      A,(DE)
3A3A 13                 INC     DE
3A3B CD DB 32           CALL    ISDIGIT
3A3E 38 13              JR      C,GETLN2
3A40 29                 ADD     HL,HL
3A41 4D                 LD      C,L
3A42 44                 LD      B,H
3A43 29                 ADD     HL,HL
3A44 29                 ADD     HL,HL
3A45 09                 ADD     HL,BC
3A46 D6 30              SUB     '0'
3A48 4F                 LD      C,A
3A49 06 00              LD      B,0
3A4B 09                 ADD     HL,BC
3A4C F1                 POP     AF
3A4D 3D                 DEC     A
3A4E 20 E8              JR      NZ,GETLN1
3A50 1A                 LD      A,(DE)
3A51 13                 INC     DE
3A52 F5                 PUSH    AF
3A53 C1         GETLN2: POP     BC
3A54 C1                 POP     BC
3A55 C9                 RET
3A56            ;
3A56            ; Get Command
3A56            ;
3A56 CD A1 3A   GETCMND:CALL    CMNDLN
3A59 3E 20              LD      A,' '
3A5B 06 23              LD      B,35
3A5D CD F4 1F   CLRLN:  CALL    PRINT
3A60 10 FB              DJNZ    CLRLN
3A62                    ;
3A62 CD A1 3A           CALL    CMNDLN
3A65 CD 80 3A           CALL    PRMES
3A68 1A         GETCM1: LD      A,(DE)
3A69 13                 INC     DE
3A6A 2C                 INC     L
3A6B FE 3A              CP      ':'
3A6D 20 F9              JR      NZ,GETCM1
3A6F E5                 PUSH    HL
3A70 CD 1E 20           CALL    CSRSET
3A73 ED 5B 76 1F        LD      DE,(KBFAD)
3A77 CD D3 1F           CALL    GETL
3A7A E1                 POP     HL
3A7B 26 00              LD      H,0
3A7D 19                 ADD     HL,DE
3A7E EB                 EX      DE,HL
3A7F C9                 RET
3A80            ;
3A80            PRMES:
3A80            ;
3A80 F5                 PUSH    AF
3A81 D5                 PUSH    DE
3A82 1A         PRMES1: LD      A,(DE)
3A83 13                 INC     DE
3A84 B7                 OR      A
3A85 28 17              JR      Z,PRMES3
3A87 FE 5E              CP      '^'
3A89 CC F4 1F           CALL    Z,PRINT
3A8C FE 20              CP      20H
3A8E 30 09              JR      NC,PRMES2
3A90 F5                 PUSH    AF
3A91 3E 5E              LD      A,'^'
3A93 CD F4 1F           CALL    PRINT
3A96 F1                 POP     AF
3A97 C6 40              ADD     A,'@'
3A99 CD F4 1F   PRMES2: CALL    PRINT
3A9C 18 E4              JR      PRMES1
3A9E D1         PRMES3: POP     DE
3A9F F1                 POP     AF
3AA0 C9                 RET
3AA1            ;
3AA1            CMNDLN:
3AA1            ;
3AA1 3A 5B 1F           LD      A,(MAXLN)
3AA4 3D                 DEC     A
3AA5 2E 00              LD      L,0             ; COL
3AA7 67                 LD      H,A             ; LINE
3AA8 C3 1E 20           JP      CSRSET
3AAB            ;
3AAB
3AAB 00 00      #TXTEND:DEFW    0
3AAD 00 00      #CURADR:DEFW    0
3AAF 00 00      #CURLN: DEFW    0
3AB1 00 00      #NXTADR:DEFW    0
3AB3 00 00      #NXTLN: DEFW    0
3AB5            UPDATEFLAG: DS  1
3AB6            LenCutBuf:  DS  2               ;Length Of Cut-buffer
3AB8            IsTxtChg:DS     1
3AB9
3AB9            SPBUF:      DS  2
3ABB
3ABB
```

▶勝手な話題ですが，我がバスケ部は区内私立６校大会でどうやら２位になりそうです。というのは自分たちの試合はすべて終わっていて，あとは相手どうしの試合結果を待っているだけなのです。それにしても２位になったら賞状も出るし，う～ん楽しみ楽しみ。

藤木　健二(16)東京都

▶ 全機種共通システムインデックス ◀

＊以下のアプリケーションは，基本システムであるS-OS“MACE”またはS-OS“SWORD”がないと動作しませんのでご注意ください。

1985

■85年６月号
序論　共通化の試み
第１部　S-OS“MACE”
第２部　Lisp-85インタプリタ
第３部　チェックサムプログラム
■85年７月号
第４部　マシン語プログラム開発入門
第５部　エディタアセンブラZEDA
第６部　デバッグツールZAID
■85年８月号
第７部　ゲーム開発パッケージBEMS
第８部　ソースジェネレータZING
■85年９月号
インタラプト　S-OS番外地
第９部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表

1986

■86年１月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年２月号
第15部　S-OS“SWORD”
第16部　Prolog-85入門(1)
■86年３月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年４月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年５月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年６月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ＆エディタ
第24部　“SWORD”2000 QD
連載　対話で学ぶmagiFORTH
特別付録　PC-8801版S-OS“SWORD”
■86年７月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS“SWORD”
■86年８月号
第26部　対局五目並べ
第27部　MZ-2500版S-OS“SWORD”
■86年９月号
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法<1>
■86年11月号
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法<2>
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC料理法<3>

1987

■87年１月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法<4>
■87年２月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアベ作成ツールCONTEX
■87年３月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　“SWORD”再掲載とMAGICの標準化
■87年４月号
第40部　INVADER GAME
第41部　TANGERINE
■87年５月号
第42部　S-OS“SWORD”変身セット
第43部　MZ-700用“SWORD”をQD対応に
■87年６月号
インタラフト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年７月号
第46部　STORY MASTER
■87年８月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS“SWORD”
■87年９月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS“SWORD”
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　X1turbo版S-OS“SWORD”
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ＆ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo版“SWORD”アフターケア
ラインフリントルーチン
特別付録　PASOPIA7版S-OS“SWORD”

1988

■88年１月号
第58部　FuzzyBASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年２月号
第59部　シューティングゲームELFES
■88年３月号
第60部　構造型コンパイラ言語SLANG
■88年４月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年５月号
第63部　シューティングゲームELFESⅡ
第64部　地底最大の作戦
■88年６月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年７月号
第67部　マルチウィンドウドライバMW-1
連載　構造化言語SLANG入門(2)
■88年８月号
第68部　マルチウィンドウエディタWINER
■88年９月号
第69部　超小型エディタTED-750
第70部　アフターケアWINERの拡張
■88年10月号
第71部　SLANG用ファイル入出力ライブラリ
第72部　シューティングゲームMANKAI
■88年11月号
第73部　シューティングゲームELFESⅣ
■88年12月号
第74部　ソースジェネレータSOURCERY

1989

■89年１月号
第75部　パズルゲームLAST ONE
第76部　ブロックゲームFLICK
■89年２月号
第77部　高速エディタアセンブラREDA
特別付録　X1版S-OS“SWORD”<再掲載>
■89年３月号
第78部　Z80用浮動小数点演算パッケージSOR
OBAN
■89年４月号
第79部　SLANG用実数演算ライブラリ
■89年５月号
第80部　ソースジェネレータRING
■89年６月号
第81部　超小型コンパイラTTC
■89年７月号
第82部　TTC用パズルゲームTICBAN
■89年８月号
第83部　CP/M用ファイルコンバータ
■89年９月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲームPUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリDIO.LIB

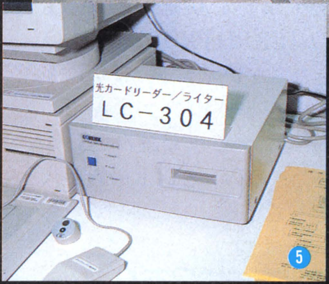
1990

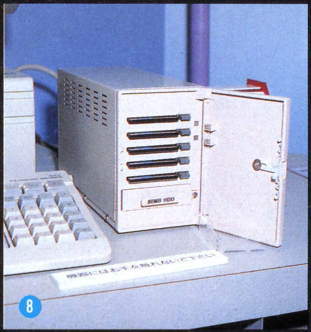
■90年１月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年２月号
第89部　超小型コンパイラTTC++
■90年３月号
第90部　超多機能アセンブラOHM-Z80
■90年４月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年５月号
第92部　インタプリタ言語STACK
■90年６月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲームSQUASH!
第95部　X68000対応S-OS“SWORD”
特別付録　PC-286対応S-OS“SWORD”
■90年７月号
第96部　リロケータブルアセンブラWZD
■90年８月号
第97部　リンカWLK
■90年９月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコード対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ

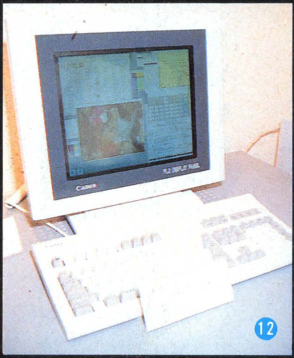
1991

■91年１月号
第102部　ブロックアクションゲームCOLUMNS
■91年２月号
第103部　ダイスゲームKISMET
■91年３月号
第104部　アクションゲームMUD BALLIN'
■91年４月号
第105部　SLANG用カードゲームDOBON
■91年５月号
第106部　実数型コンパイラ言語REAL
■91年６月号
第107部　Small-C処理系の移植
■91年７月号
第108部　REALソースリスト編
■91年８月号
第109部　Small-Cライブラリの移植
■91年９月号
第110部　SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部　Small-C活用講座（初級編）
■91年11月号
第112部　Small-C活用講座（応用編）
第113部　MORTAL
■91年12月号
第114部　Small-C SLANGコンパチ関数

data データショウ show







1 シャープの高速5.25インチMOドライブ
2 川崎製鉄の高速光ディスク
3 21Mバイトのフロプティカルディスク
4 本当に小さい1.3インチハードディスク
5 磁気カードと同じ大きさで，2.8Mバイトの容量を持つ追記型光カードリーダ/ライタ
6 見えないステルス型バーコード
7 最大5.2Gバイトのディスクアレイ
8 メモリカードを使ったICディスク
9 99,800円のカラーインクジェットプリンタ
10 VRで体験する旋盤シミュレーション
11 27枚格納できる3.5インチMOオートローダ
12 キヤノンのFLCディスプレイ
13 FTAM規格によるファイル転送の実演

9月16日から19日まで，東京国際見本市会場でデータショウ'92が開催された。

今回のデータショウでは，まず大容量デバイス関連を中心にまとめてみた。はっきりいってそれ以外ほとんど見所はない。

川崎製鉄ではハードディスクをバッファとして使い，光磁気ドライブの高速化を図っていた。コマ数は粗いが（秒間5コマぐらいか？），5.25インチ光磁気ディスクドライブから取り込み映像を垂れ流せる転送速度を実現している。同じようなもので，エムアイシー・アソシエーツでは，3.5インチ光磁気ディスクに880Mバイトのハードディスクを載せていた。

リムーバブルメディアとしての筆頭は光磁気ディスクだが，フロプティカルディスク，小型リムーバブルハードディスクにも注目したい。フロプティカルディスクは，ドライブ自体の速度の問題や国内生産がまだ行われていないということもあり，これからが期待される。

ハードディスクでは，ノート型コンピュータの普及に伴ってか小型化が進んでいる。代表的なのがヒューレット・パッカートの1.3インチハードディスク。まさに健気に動いている，という形容がぴったりだ。

そして，ちょっと気になる新製品として同社製の300dpiカラーインクジェットプリンタ。定価99,800円は，パーソナルユースとしてなかなかの魅力がある。

そのほか，ネットワーク化によるデータの有機的統合など，データのマルチ展開を目指したデモが各所で実演されていた。

全体的に見ると，目新しさよりも着実な技術の進歩を間近に見ることができ，これからを期待させてくれる。

新製品紹介

CHART PRO-68K

Kaneko Shunichi 金子　俊一

数少ないX68000用ビジネスソフトのラインナップに，ビジネスグラフチャートソフト「CHART PRO-68K」が加わった。データベースで作成したデータをもとに，折れ線，立体など，自分の好きなかたちのグラフを描く。プライベートでも使ってみたい？

このソフトはウィンドウシステム上で動作する。といっても，SX-WINDOWではない。もちろんMS-Windowsでもない。どうやら，「CARD PRO-68K」と同じウィンドウシステムのようだ。開発開始時期などの問題ではあろうが，SX-WINDOW用でないのはやはり残念である。

このシステムでは最大12枚のウィンドウが開ける。実際に使ってみた感じでは必要十分な枚数だといえよう。SX-WINDOW ver.2.0のような画面スクロールはしないので，あまり開きすぎると画面の中がゴチャゴチャしすぎるからだ。ウィンドウシステムという以上，スクロールバー，スクロールボックス，ズーム，リサイズボックスなどはすべてサポートしている。操作は一般的なクリックやドラッグなどで行われる。このあたりはマニュアルを読まなくても十分に対応できる。それなりによくできたシステムである。メモリが4Mバイト以上になっている人はウィンドウシステムの背景を好きなグラフィックにすることもできる。

問題点をあげるなら，ウィンドウ内の仮想画面が広すぎてスクロールバーが事実上役立っていないこと。4096×999項目も扱えるのに，1画面に表示されている範囲はせいぜい20×20項目程度。全体の0.01%にも満たない。10,000画面分もデータがあるのだから，スクロールバーを使ったらアサッテのあたりを表示してしまう。

X68000用 3.5/5″2HD版3枚組 38,000円(税別)
シャープ ☎03(3260)1161

まずはデータ編集

さて，このCHART PRO-68Kには大きく分けて3つのウィンドウがある。1つはデータを編集するデータ編集ウィンドウ。ストレートな呼び方が好感をもてる。さらにグラフ編集ウィンドウがある。そのほかのウィンドウは電卓とかFEPなどで，どちらかといえばシステム寄り。

グラフを作るためにはまずデータから，というわけでデータウィンドウを説明していこう。先ほどもふれたとおり，項目数は4096行×999列。個人で使用するレベルならば十分といってよいだろう。ちょっと思い浮かべても，1000項目ものデータをいじるのは考えにくい。かえって項目数が設定できて，スクロールバーなどを有効に利用するほうが使い勝手はよかっただろう。

肝心のデータ入力はキーボードで行う。行はR1からR4096まで，列はC1からC999までである。縦長の構成といえる。データはそれぞれのマス目にひとつずつ入ることになり，場所の指定はR4C2などというふうに表す（図1）。

CHART PRO-68Kはグラフ作成ソフトと述べたが，グラフを描くだけがすべてではない。充実した演算や分析機能があり，その結果をグラフにできると考えたほうがよいだろう。統計では総和，総2乗和，最大値，最小値，平均などを求めることができ，ほかにも標準偏差やら度数分布やらと，確率統計学で習ったことがバシバシと出てくる。「まじめに勉強しておいてよかった」と思える至福の瞬間である。

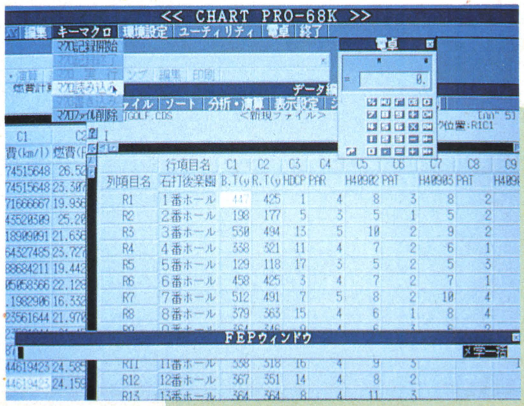
もちろん，データは入力するだけがすべてではない。過去のデータを有効に利用できることも重要である。「CHART PRO-68K」では「CARD PRO ver.2.0」のデータを読み込むことができる。ほかにも拡張子が.K3のCSV形式のファイルや，SLKのSYLK形式ファイルも読み込むことができる。また，CSV，SYLKの両形式で出力することもできる。

範囲指定方法に？あり

データを入力/作成/編集するならば，カット&ペーストでもソートでも範囲を指定する必要がある。また，豊富な演算や分析を行うにも範囲指定は欠かせない。やり方はマウス左ボタンでのドラッグなので感覚的には理解しやすい。ただし，1画面内にデータがあれば，である。

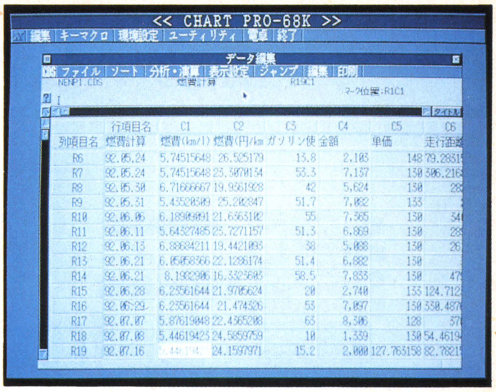
たとえば，演算の処理でR1C1からR30C1を足して，R31C1に入れたいときがあったとする。この事例は比較的ありがちで，実際にも今回作ったゴルフのスコアの例では，R1C5からR18C5を足してトータルスコアを求めている。

Rは行なので縦表示であり，画面サイズを最大にしてもR1とR30では1画面内には入りきらない。そこでR1からドラッグしたまま画面下端までマウスを引っ張っても，



例のウィンドウシステム。電卓も出る

グラフに注釈をつけることも可能



行列の位置はRI, CIというふうに表す

画面はスクロールしてくれない。つまり，ドラッグで範囲指定できるのは1画面以内というマヌケな事態が発生している。

キーボード操作ではカーソル下を押しつづければスクロールするので，範囲指定はできるのだが，ウィンドウシステムとしてちょっと納得できないものがある。

救いというわけでもないが，キーボードマクロが使用できる。定義したマクロはもちろん保存することができる。できれば入力以外の操作はマウスオンリーでできるのが楽ちんだと思うのだが。

そしてグラフ編集

グラフの編集は人間的である。まず紙の大きさを選び，その中でこのへんをグラフにしようとか，ここに注釈を入れよう，といった感覚で作業を進めることができるのだ。プリントアウト感覚はわかりやすく，DTPの基礎ともいえそうだ。

また，扱えるグラフの数も過去のX68000用のソフトをぶっちぎっている。とりあえずは写真を見ていただきたい。あれもグラフ，これもグラフといった感じである。君はこれらのほかにグラフを思いつけるか？

2次元のグラフが15種類，3次元のグラフが10種類，このほかにユーザーがカスタマイズしたグラフも使うことができる。表示範囲の最大値や最小値を設定したり，原点の値を設定するなどは当たり前。データがあまりにもかけ離れた数値のときなどに使う途中省略や目盛間隔を設定することもできるし，対数目盛にすることもできる。3次元表示では角度やら視点の位置を変えることができる。

もちろんグラフ上に注釈を入れることもお茶の子サイサイである。文字は注釈用に準備したスペースに合わせて，自動的に文字サイズが決められるようだ。

「グラフだけでは説得力がない。表も一緒に載せたい」

うむうむ，当然の欲求であろう。グラフ編集ウィンドウに表を書くのは非常に簡単な作業である。

まずはデータ編集ウィンドウ上で表にしたいデータを範囲指定する。そこで「編集」をドラッグして「コピー」を選ぶ。これでデータがバッファに入った。次にグラフ編集ウィンドウをアクティブにして，「グラフ」をドラッグ，「表作成」を選んで左上と右下をクリック。サイズが決まると自動的に表は出来上がっている。もちろん，サイズによって文字の大きさが変わる。

実はドロー系のツール

実はこの「CHART PRO-68K」はドロー系のツールである。1つひとつのグラフィックデータを部品として扱っているのだ。データをあとから編集するのも思いのままである。ドロー系といえば「CANVAS PRO-68K」を思い浮かべる人が多いだろう。皆さんの期待を裏切るようでもうしわけないが，データを「CANVAS PRO-68K」に持ち込むことはできない。逆もまた不可。

そうはいっても「PressConductor PRO-68K」の.GPCファイル，「Multiword」の.MGRファイル，SX-WINDOWの.PIXファイル，X-BASICの.GS3ファイル，「Z'sSTAFF」の.ZIMファイルにすることができるので，利用方法はかなり広がるだろう。特に「Multiword」との組み合わせはかなり強力なコンビになりそうである。

また，グラフ編集ウィンドウの中に「任意作図」というものがあり，線を引いたり，円を描いたりといった程度のことはできる。また，任意の大きさの文字を書くことも可能である。ドロー系としてはごく基本的なものしかないが，ちょっとした絵なら描くことも可能である。部品と呼ばれるドローイングデータも各都道府県の形状を中心に，86種類が用意されている。

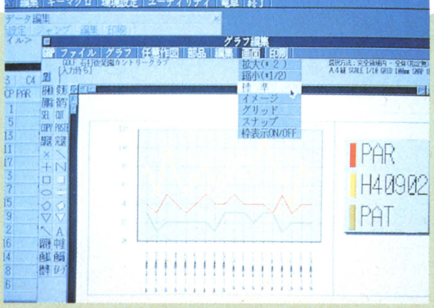
値段がネックか？

もちろん，ハードディスクにインストールできる。専用の実行ファイルが付属してくるので，インストールは初心者でも楽にできるだろう。

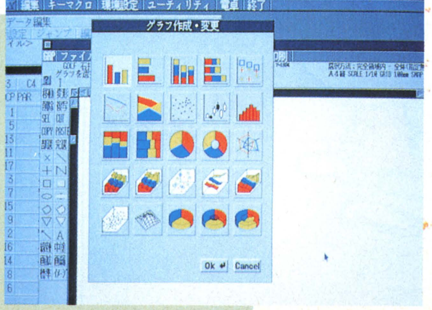
また，FIXER対応というか，マニュアルにきっちりとFIXER使用上の注意点（コンフィグファイルについて）が記載されている点も見逃せない。あまり大量の文章を書くわけではないので，FEPはなんでもよさそうだが，使い慣れたものがいちばんであろうから，FIXER愛好者にも安心だろう。

最初に使ってみましょう風の解説が載っているマニュアルはわかりやすかった。これならこのテのソフトにうとい人でも明日はホームランだろう。

データ編集に若干の難があったが，キーボードユースではそれすら問題ではなくなる。10MHzでは心持ち遅いが，機能からすればそこそこといったところか。やはり問題は価格設定であろう。ビジネスソフトとしては安いほうかもしれないが，X68000のユーザー層を考慮に入れて，一般ユーザーにも手が届くような価格にすべきではなかっただろうか。一般市民もグラフは作りたいけど，ちょっと具が大きいぞ。



入力したゴルフの成績を折れ線グラフに



グラフの種類はわりと豊富

［第18回］
風の吹いた日

寺尾　響子

響子 in CGわ〜るど

　青い風がことのほか高く感じられる，風の強い日でした。休日のビジネス街は閑散としていて，ふだんとは別の場所のように思えます。冷たいコンクリートとガラスにはさまれた，風の通り道をひとりで歩いてゆきます。

　高いビルとビルのはざまに小さな空き地が見えてきました。

　誰かがいる？

　空き地の中央に，髪はほとんど真っ白なのに顔はまだ14,5歳の少年のような人が，四角く大きな黒い石に向かって座っていました。石は2001年宇宙の旅に出てくるモノリスのようにも見えます。

　近づいてゆくと，その人は椅子から立ち上がって，30度の角度で腰を曲げてきちんとあいさつをしました。

「僕は芸術家です。あれ，それとも芸術家だったのかな，芸術家になろうとしているのかな，忘れてしまった。……でも，とにかく，ほら」

と大きく腕を広げました。

「こんなにたくさんの道具を持っている。何百とある。ノミやかなづちはもちろん，充電式ドリルやコンピュータ制御の工作機械だってあるぞ。なのに，さっぱりだ。全然，手がつけられないんだよ。どうしてだろう」

「どうしてでしょうねえ……」

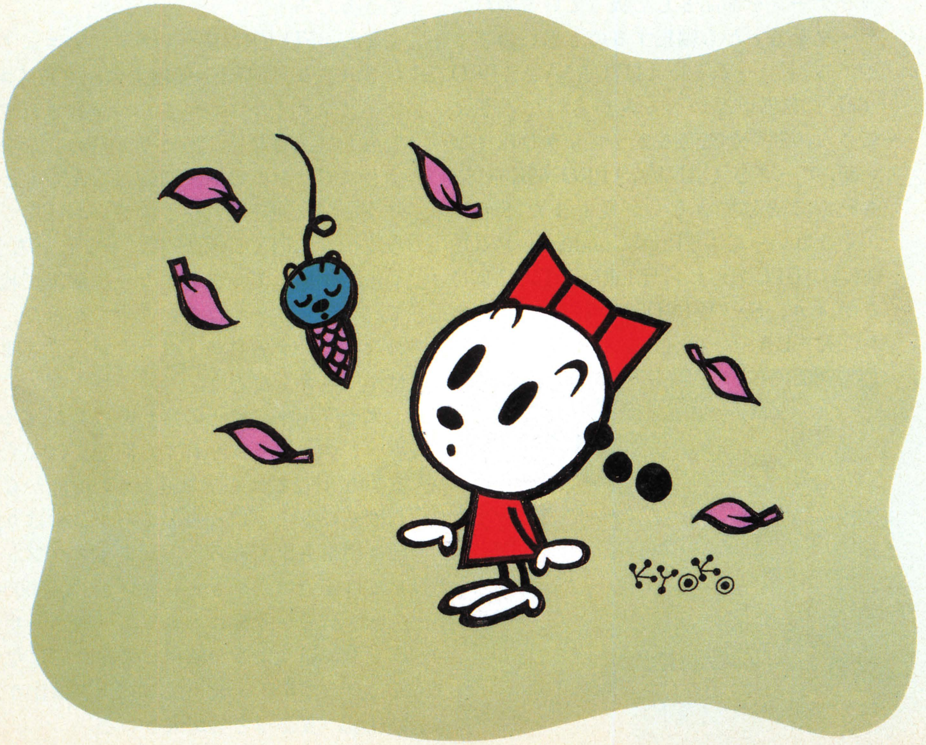
　答えが一瞬浮かびかけましたが，頭の中をぴゅうと小さな風が吹き抜けて，彼方へ飛ばしてしまいました。

　くるくる，くるる。

必要なもの

　道具があるのに作れないのは，とても苦しいし，つらい。

　そういえば，私だって絵を描くための道具はた



くさん持っています。筆，ペン，色鉛筆，エアブラシのコンプレッサ，スクリーントーン，カラーインク，水彩絵の具に，アクリル絵の具，イラストボード，そしてX68000。どれもこれも今は必要に思えるものばかりだけれども。

いきなり大きな風がやってきて，空き地にあったすべてのものをぶわんと持ち上げました。芸術家らしい人はおおあわてで，できるだけ多くの道具を抱え込もうとしました。が，風の力はそれよりも強く，道具もろとも，その人をぴゅうと吹き飛ばしました。

風はビルのガラス窓に当たって跳ね返り，今度は私の体をぶわっと運び上げました。

スモッグのない澄み切った東京の空に，芸術家らしい人と私は木の葉のように舞い上がり，輪を描きました。

くるくる，くるる。

輪がだんだんと広がってゆきます。

「わかったぞ。何も持っていないということが，僕にとってはいちばん必要な道具なのだ」

と叫びながら，その人は小さな虫のようになって，遠くへ消えてゆきました。

THE USER'S WORKS

●SX福袋

今回紹介するのはX68000USER'S CIRCLE KISARAGIが制作したSX-WINDOW用のアクセサリ集だ。これまで作りためていたものをまとめたものだそうだ。

ついに同人ソフトにもSX-WINDOW版が現れるようになったのだ（しみじみ）。

●SXEYES

いわゆる「目玉」である。X-WINDOW風の目玉型ウィンドウを実現している。もちろん標準のSX-WINDOWではこのような形のウィンドウを用意していないので，「自前でウィンドウ定義関数を書く」という荒技を使っている。アイコン化しても動く目玉がなかなか可愛い。

●LINE CRASH！

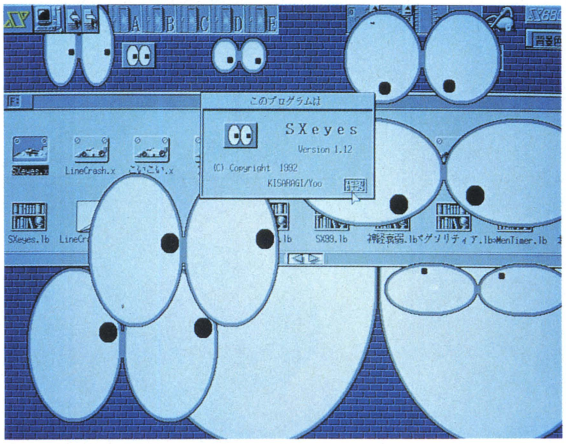
懐かしのライトサイクルゲームのSX-WINDOW版。2人対戦専用で，ジョイスティック2本が必要となる。

●こいこい

美しいグラフィックデータに注目！　ちょっと見た目にはテキストだけで描かれたものには思えない。もちろんゲームもまっとうに遊べる。

●定規

画面上のドット数を測るためのユーティリティ(？)。SX-WINDOW用ソフトの習作として作成されたものだそうだ。一応，実用にもなる。



目玉の競演。あまり多くすると重い……

●4D MAZE

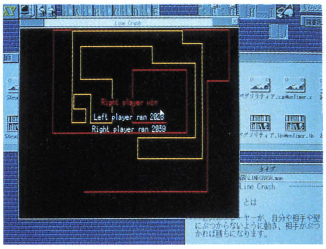
2次元マップ2個を頼りに出口を探してさまよう4次元迷路。これも習作。キャラクタで描かれたマップがちょっと悲しい。

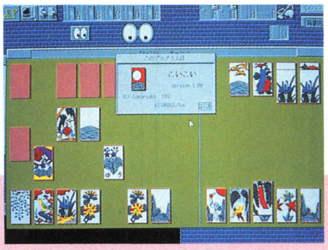
●ペグソリティア

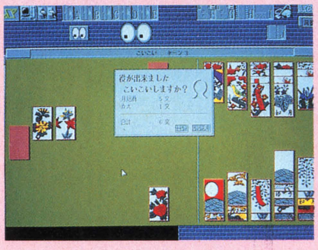
その名のとおりペグソリティア。一応解説すると，ひとつ飛ばしにピン（ここでは球）を動かし，飛び越したピンを抜いていく。最後の1本になるまで続けるというパズルゲーム。

●タイムアタック神経衰弱

花札を用いたひとり神経衰弱。どれだけ早くすべての札をあわせられるかを競うというもの。スコアがセーブされないのがやや残念。

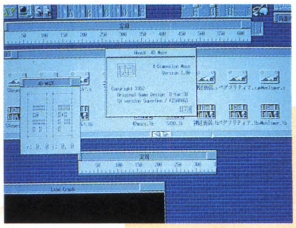

LINE CRASH！

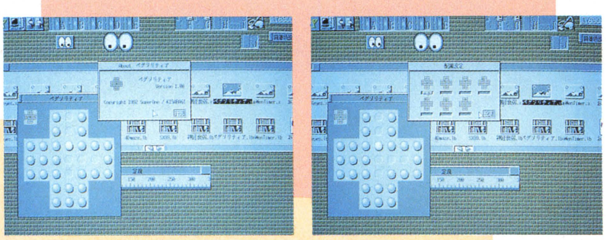



花札こいこいのSX-WINDOW版。データはリソース化されている

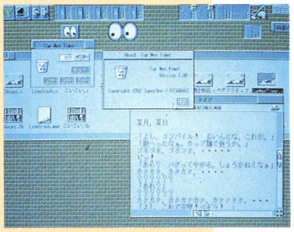
●カップ麺タイマ

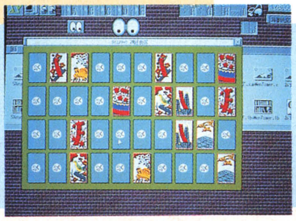
1分単位のアラームを設定するタイマ。アラームをセットしておけば，SX-WINDOWで作業していてもカップラーメンを作り損なうことがない(？)。実用性は不明。


定規と4D　MAZE


ペグソリティア。初期パターンもいろいろ

ちょっとした息抜きに最適のひとり神経衰弱


カップ麺タイマ



入手方法

●郵便振替の場合

口座番号　東京8-609033

口座名称　KISARAGI

上記の口座に2,000円を振り込み，通信欄にはメディアの種別（5インチまたは3.5インチ）を明記すること。

●郵便小為替の場合

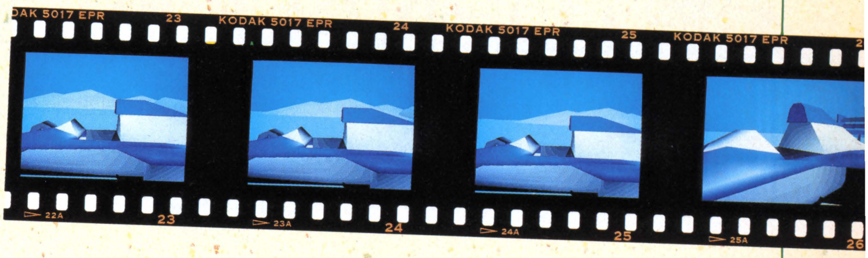
郵便小為替2,000円分（無記名）を同封し，メディアの種別ほか必要な情報を記入して封書で下記の住所まで連絡する。なお，事故防止のため，返送先住所氏名を記入したタックシールをつけることが望ましい。

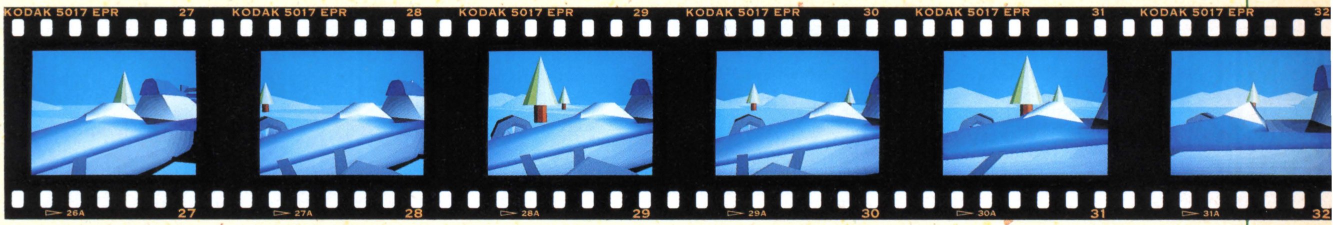
〒191　東京都日野市三沢570-1
ハイツうわた206号
今川方 KISARAGI

Oh!X Graphic Gallery

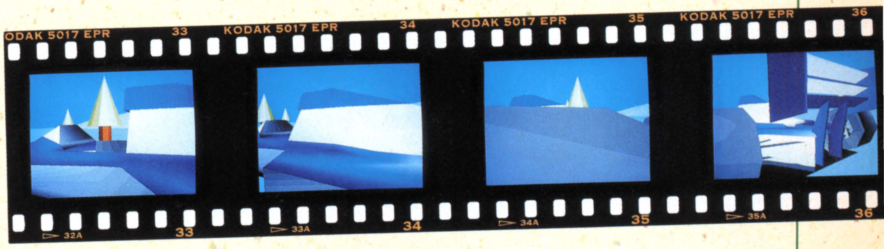
DōGA CGアニメーション講座

今回も基本的にはF1を動かしているだけですが，前回よりも，視点，画角，フレーム間のつながりなどをより効果的に設定していきます。前回同様，このカラー記事を参照しながら，連載記事中のアニメーションファイルを作成してください。

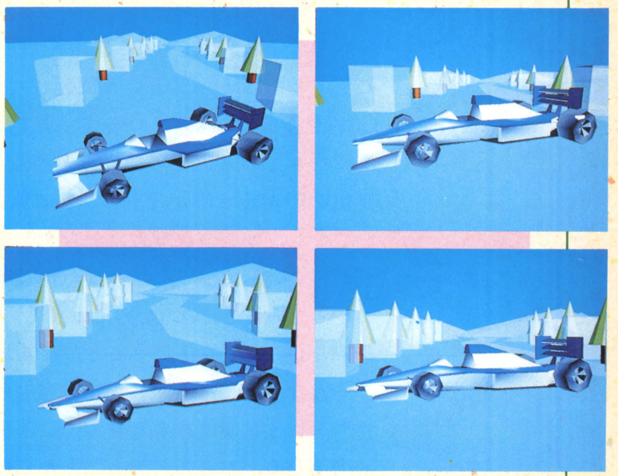




（4フレームおきに撮影）背景が遠くにある山々だけではスピード感が出にくいので，道の両側に木を植え，さらに，カメラを追い越すような動きを設定します。



（1フレームおきに撮影）オーバーラップを使って，カットごとのつながりの不自然さを解消させます。でも，あまり多用しないこと。よく考えたうえで。



左上：広画角／ハイポジション，右上：広画角／ローポジション，左下：狭画角／ハイポジション，右下：狭画角／ローポジション

打倒TORNADOへの第一歩(後編)

プロジェクトチームDōGA
かまた ゆたか

今回は，CGAを始めたばかりの人が行き詰まる初歩的な問題点を分析し，具体的な解決方法を検討します。これを修得すれば，オリジナルCGA作品の完成に一歩近づくこと間違いなし！

はじめに

私の趣味のひとつに海外旅行があります。というと，なんかとってもリッチに聞こえますが，いわゆるバックパッカーというやつで，リュックサックひとつで各地を放浪します。

今年もインドかトルコ，エジプトあたりに行こうかなと思って，ガイドブックなど読んでいるところです（連載は1回休み？)。夕焼けをバックにピラミッドがシルエットを作る写真なんかを見て，ふと思いました。

“あっ，アンチエリアスしていない……”

さらに，新都庁を初めて見たとき“面数が多い”と思った，のは私だけか？

さて，この連載では，“TORNADOのような作品は誰にでもできるか？”という問題に対して，実際にF1のCGAを作ってもらっています。まず，モーションデザインツールであるFFEの使い方を勉強しました。これで，野山をF1が走るCGAを作る方法は修得できたはずです。

しかし，使い方を覚えたからといって，すぐにみんながかっこいいカットを作れるわけではありません。実際にやってみると，いろいろとうまくいかず，“オレには才能がないんだー”ということになりかねません。

そんなふうにうまくいかない理由を“センス”や“感性”という言葉で終わらせるのには賛成しかねます。初心者の場合，むしろ“努力”とか“経験”の問題だと思います。大部分のうまくいかない理由には，ちゃんと原因があり，論理的に説明でき，解決できるでしょう。

もちろん，この誌面上ですべての原因を説明するスペースはありませんし，私自身，そんな力量はありませんので，比較的よくありがちな原因と，その解決法について紹介したいと思います。

かっこいいカットが浮かばない

症状：FFEの使い方は修得し，すべての位置座標を指示されればCGAは作れるが，いざ自分でかっこいいカットを作ろうとすると，どんなカットを作ればいいかアイデアが出てこない。2,3の作品は作ってみたが，かっこよくないし，この程度ならお試しシステムでもできてしまう。

こういう症状に思い当たる人は結構多いのではないですか。そして“やっぱりオレにはセンスが……”と思っているのでしょうか？

ご安心ください。この問題は，案外簡単に解決します。ずばり，人のマネをしましょう！

こう書いてしまうと，オリジナリティがどうのという人がいますが，そんなことは気にする必要はありません。だいたい，どんなカットを作ったらいいかアイデアも浮かばないような人が，オリジナリティを口にするなどおこがましい。漢字の書き取りもできない子供が書道の真髄を語るようなものです。そんなことを気にするのはCGAコンテストで活躍できるようになってからで十分です。まずは，基礎を身につけましょう。

初心者が人のマネをすることは，決して恥ずかしいことではありません。何事においても，最初は人マネから入っていくものです。それどころか，一流になっても人マネは有効です。リチャード・ドナー監督が，「リーサル・ウェポン」を制作するにあたっては，いままでのアクション映画の名作を片っ端から見て，その面白いエッセンスを集めていたといいきっているぐらいです。

さて，今回はF1のCGAを作っていますので，近所のレンタルビデオ屋さんにでも行って，車のプロモーションビデオ，車のCM集，××年度F1グランプリ名シーン集などを借りてください。カーアクションを売り物にしている映画だってかまいません。

それらを数本借りてきて，かっこいいカットを探します。もし可能ならば，かっこいいと思ったカットばかりを集めて，アイデア集のビデオを作るのもよいでしょう。紙に絵コンテ風に記録しても結構です。これだけで，10や20カットのアイデアが溜まるでしょう。

次にそのアイデアの中で気に入った，なおかつ簡単そうなカットを実際に作ってみましょう（車が爆発炎上なんていう難しいカットは決して選ばないでください)。

カットをマネして作るとき，注意しなければいけないのが，カメラの設定です。車自身の動きは画面を見れば一目瞭然ですが，カメラの位置，画角などはよく見なけ

ればわかりません。とりあえず作ってみたが雰囲気が全然違うというときは，カメラの高さや画角が違うケースが多くあります。Oh!X Glaphic Galleryの写真も参考にしてください。

このようにして，人のマネをして数カット作ってみると，それぞれのカットにいろんな工夫がしてあることに気がつくでしょう。カメラの画角を極端に小さくして遠近感をなくしていたり，視線を軸にゆっくり回転(rotx)しながらパンしているといったカメラワーク。車を画面の中央からわざと外したり，ほかの物体と組み合わせる構図的な問題。画面を分割したり，テロップを入れるなどのエフェクト。また，CGAシステムでは表現できないところ（炎や煙）にかっこよさがあるかもしれません。

人マネから，そういったかっこよさの理由を見つけていけば，今度はそれらを組み合わせた新しいアイデアが膨らんでくるでしょう。そうなれば，あとは実行あるのみです。そしてまたアイデアに詰まったら，積極的に人の作品のよいところを盗んでください。

FFEで思うように動かない

症状：たとえば，コーナーを曲がって走り去るとか，S字に曲がるカットを作ろうとしたのだが，思ったような動きにならず，グネグネと動いたり，とんでもないところに行く。設定するフレーム数を増やして，丁寧に指定しているのに，ぜんぜん直らない。

まず，FFEの使い方の基本として，“設定するフレームは3つ以下”と覚えておいてください。

物体（またはカメラ）が直線的に動くときは，最初の位置と最後の位置の2カ所を指定すればよいのはご存じのとおりです。曲がって動くときは，最初の位置，最後の位置，そしてその真ん中ぐらいの位置を指定します(図1)。それ以外のカットは，FFEでは基本的にできないぐらいに考えてください。

では，S字コーナーのように，左右逆のカーブを曲がったり，曲がったあとで直線的に走り去るようなカットはどうするのかといえば，複数のカットに分割するのです（図2，3)。こうすれば，どのようなカットも直線か，単純なひとつのカーブになります。

なぜこのように，わざわざ複数のカットに分割するかといえば，FFEで補間に用いているスプラインに原因があります。このアルゴリズムは，人間が通常考えるような滑らかさとはちょっと違うのです。たとえば，4つの点をつなげるとき，人は普通，図4－1のように，cとdの間を直線で結びます。しかし，コンピュータはすべての点を滑らかな曲線でつなげようとするため，図4－2のようになってしまいます。

図4ではまだましですが，点bとcとの間隔が広くなると，だんだん変なつなぎ方になります(図5)。この場合，cとdの間を滑らかにしようとして，その中間にも

図1　FFEでできる動き

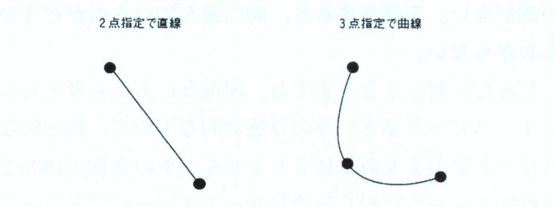


図2　直線部と曲線部は別カットに

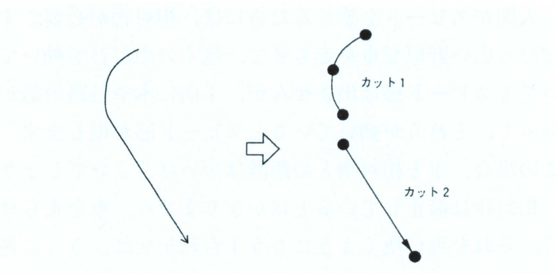


図3　複雑な曲線は単純な曲線の組み合わせに

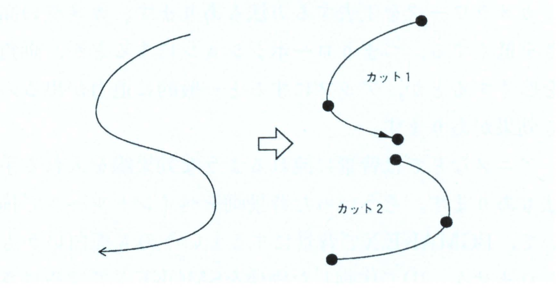


図4　スプラインによるつなぎ方

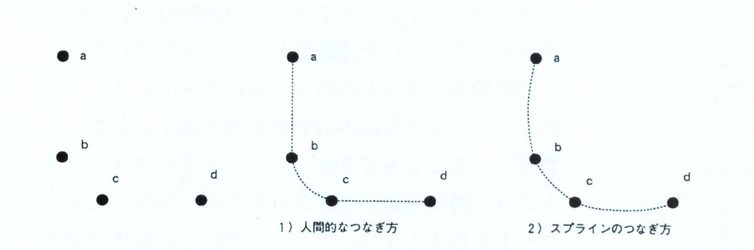


図5　極端な例

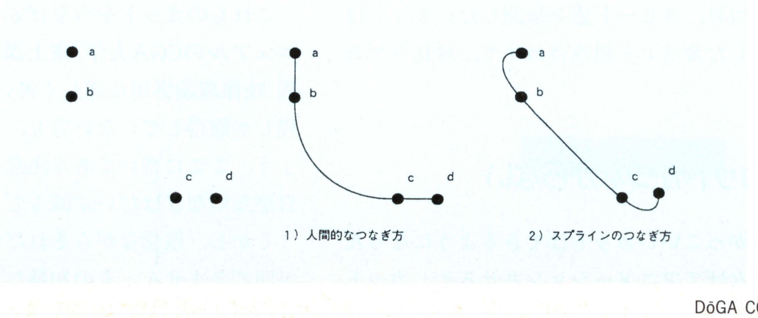
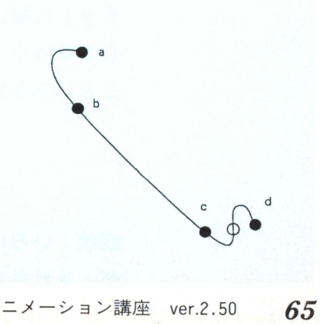


図6　グニャグニャ



う1点新たに設けると，図6のように，さらにグニャグニャしてきます。このグニャグニャは指定する点の数が多く，点と点との距離が近いほど激しくなります。

スピード感が出ない

症状：1カット作ってみたのはよいが，ぜんぜんスピード感がない。下手をすると，前に進んでいるのかどうかもわからない。

だんだん難しくなりますね。理由はいろいろ考えられます。スピード感というのは感覚的なもので，物理的なスピードはあまりあてになりません。車の全長が1000で，毎秒20フレームだから時速300kmのスピードだといっても，そう見えなければ意味がありません。

人間がスピードを感じるためには，相対物が必要です。だだっ広い野原で車を走らせて，遠くの山だけが動いていてもスピード感は出ませんが，手前に木や道路標識があって，それらが動いているとスピード感が増します。この場合，車と相対物との距離は短いほどよいでしょう。

相対物は静止しているとはかぎりません。車を走らせて，それを追い抜くようにもう1台動かせば，さらに速度感が増します。同様に，動いているカメラを追い抜くようにするのもひとつの手です。

カメラワークを工夫する方法もあります。カメラの高さを低くする，つまりローポジションにするとか，画角を広くするとか，アップにすると一般的に迫力が出るので効果があります。

アニメなどでは背景に流れるような効果線を入れる手法もあります。そういった背景画をペイントツールで描いて，BGMAKE.Xで背景にするというのも面白いかもしれません。3Dで作画した画像をSMOKE.Xでぼやけさせるのも少しは効果があります。また，編集によってスピード感を出すこともできます。時間的に短いカット（フラッシュカット）を連続させるという方法です。

比較的長いカットのあとに短いカットを2つ，3つ入れるというような組み合わせを繰り返すことで，リズム感をつけることもできます。このようなフラッシュカットでは一瞬の間延びも許されません。作画してタイムチャートを作るときに，フレーム単位での編集が必要です。

スピード感ある演出といえば，「EPA2ビデオマニュアル」の森山さんが有名ですが，あのスピード感のコツは，カットに緩急をつけ，スピード感を強調したいカットは必ずゆっくりとしたカットと組み合わせて，対比させることだそうです。

カットがつながらない

症状：いろいろかっこいいカットはできるようになったが，それらをつなげてアニメーションさせると，カットどうしがばらばらで，つなぎ目が不自然になってしまう。なんかしっくりいかない。

詳しく解説する前に，どういうつなぎが自然でどういうつなぎが不自然なのか，実際に見ていただきましょう。実はこの解説をするために，前回その例となるカットを作っておいたのです。

10月号では「X10A.FSC」「X10B.FSC」「X10C.FSC」という3つのカットを作りました。実は「X10C.FSC」だけは，たまたま遊びにきていた宇宙人森山さん（「EPA2ビデオマニュアル」の作者）が制作してくださったカットです。あのスピード感はさすがですね。しかし，そのカット自身は素晴らしくても，すでに作っていた「X10A.FSC」とはなんともつながりません。そこで，「X10B.FSC」というカットを作って，2つのカットを自然につなげることにしたのです。

ですから，タイムチャート「X10A.TCH」を変更して，「X10B [1-5]」を削除して「X10A」と「X10C」をつなげると不自然なつなぎになりますので，ぜひ一度見比べてください。「X10B.FSC」がたった5フレームながら重要なカットだ，といった意味がよくわかるでしょう。

なぜ，全然つながらない2つのカットが，「X10B.FSC」というカットを挿入することでつながるようになったのでしょう。ここで使われている主なテクニックは，“方向性の連続”と“モーションつなぎ”です。

“方向性の連続”とは，連続するカットにおいて，注目している物体の画面上の動きを同じように統一しておくことです。「X10B.FSC」では「TYRL.SUF」が画面上の右から左方向に動き出しています。「X10C.FSC」の出だしの一瞬で画面の右から左に横切ります。どちらかといえば，画面の手前から奥の方向に移動している点でも方向性がとれています。それに対して，「X10A.FSC」では「TYRL.SUF」は動いておらず，特に方向性は明確ではありません。

“モーションつなぎ”とは，あるアクションの最中にカットを変えることで，サイズの異なる2つのカットをつなげることです。この例では，「X10B.FSC」で始まった“右に曲がりながら加速する”というアクションにおいて，最初の1/3ぐらいまでを「X10B.FSC」で描き，残りの2/3を「X10C.FSC」で描いています。そして，サイズもフルショットからアップ（バストショット？）に切り替えています。

これらのカットをつなげるテクニックについては，マニュアルのCGA大学/修士課程/映像理論概論と博士課程/映像理論応用に詳しく書いたつもりです。まだ専門課程しか修得していない方も，一度読んでみるとよいでしょう。ここに書いてある注意やテクニックを使えば，不自然なつなぎはだいぶ減るでしょう。

しかし，残念ながらそれだけでは，すべてのカットを説明できません。その知識だけでは，自由自在に編集で

きるというわけにいかないのです。なにが足らないのか。それは“センス”，とはいいたくありません。きっと，単に私の知識や経験が少ないだけだと思います。そのテの専門書には完璧な理論が載っていると信じています。

しかしながら，実際問題として，マニュアルの知識の応用だけでは解決できないようなカットの場合はどうすればよいのでしょう。最後の手段をそっとお伝えしましょう。どうしてもつながらなければ，オーバーラップしてください。はっきりいって，これは邪道です。“ラーメンのおいしい作り方”という本に“カップラーメンを買ってくる”と書くようなものです。でも，結構使えます。オーバーラップの作り方はあとで解説します。

オーバーラップと同様の効果を出す方法として，フェードアウト，フェードインを高速に行うのも有効です。フェードアウトは，ペイントソフトなどで真っ黒の画面を用意して，それとオーバーラップさせればよいのですが，このとき真っ白の画面を使ってホワイトアウトすることもできます。CGAの場合，もっと自由にレッドアウトとか，ブルーアウトとかしても面白いかもしれません(やったことはありませんが)。

試しに1カット制作する

理屈をひととおり解説したところで，実際に先月号の続きのカットを作ってみましょう。

先月号の最後のカット「X10C.FSC」は，ロングで終わっていましたから，次のカット(X11A.FSC)は，アップぎみにします。あまりスピード感ばかり出すのも単調なので，次のカットは，ちょっとゆっくりめの安定感のあるカットにして，その次のカットをフラッシュカットにすることにしましょう。

○背景の設定

概要：まずFFEを起動し，前回と同様に背景を設定します。背景だけを設定したファイル「BACK.FSC」をちゃんと作っていた方は，それを「6) ファイル」の「LOAD」で読み込むだけで結構です。

自分でやる場合は以下のように。操作がわからない方は，先月号を復習してください。

操作：1)　背景を水色に設定

・「背景設定」の「べたぬり」でRGB＝(0.2 0.4 0.8)

2)　「JIMEN.SUF」の設定

・「物体設定」の「追加」で「JIMEN.SUF」を選択したあと，X，Y，Z方向に10倍して，Z軸方向に－70移動する

○1フレーム目の設定

概要：「TYRL.SUF」はY軸上を3500から－3000まで移動します（図7）ので，あらかじめテンキーの「＝」を2回押して表示範囲を広くしておいてください。

操作：1)　「TYRL.SUF」の設定

・「物体設定」の「追加」で「TYRL.SUF」を指定する

・位置座標を(0,3500,0)にし，Z軸回りに－90度回転して「決定」

2)　視点の設定

・「視点設定」で視点の座標を(400,3400,140)にする

・注目点の座標を(0,3400,100)として「決定」

○50フレーム目の設定

操作：1)　フレームナンバーの変更

・「フレームNo設定」でフレームナンバー「50」を入力して，リターンで決定する

2)　「TYRL.SUF」の変更

・「物体設定」の「変更」で「選択」を2回クリックして，「TYRL」を選択して「決定」

・座標入力状態になってから，位置座標はYだけ変わって，(0，－3000，0)で「決定」

3)　視点の変更

・「視点設定」で視点を(400,－3100,140)，注目点を(0,－3100，100)として「決定」

4)　セーブ終了

・「ファイル」の「SAVE」で「フレームソース」を選択し，出力ファイル名「X11A」をキー入力する

・「終了」でFFEを終わる

○作画

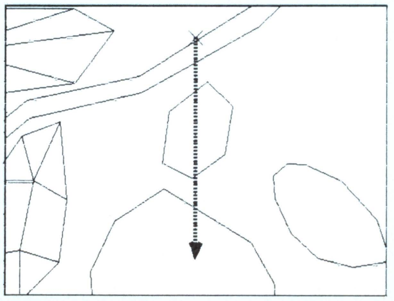
概要：作業をさせたあとでいうのもなんですが，これをこのまま作画させても，スピード感のないつまらないカットができます。つまり，悪い例です。のちほど修正しますが，まずは勉強だと思って悪い例を見てみるのもよいでしょう。ただ，画質を悪くしても作画には時間がかかりますので，時間のない方はWIREVIEWで見るだけにして，Oh!Xでも読んでるから待たされてもいいやという方はAUTOで作画してください。

操作：1)　WIREVIEWで見るだけの場合

・コマンドラインから「FF X11A」と入力し，FFでフレームファイルを作る

・コマンドラインから「WIREVIEW TYRL.

図7 「X11A」の動き



SUF JIMEN.SUF X11A.FRM /V」と入力する
・アニメーションを見て，納得すれば「ESC」を押して終了

2) 作画させる場合
・コマンドラインから「AUTO X11A.FSC」と入力する
・1)を実行していると，途中で「すでに，X11A.FRMがあります……」というメッセージが出ることがあるが，気にせず「1」を選択する

「X11A.FSC」を修正する

「X11A.FSC」にスピード感がない最大の理由は，相対物が遠くの野山しかない点です。また，カメラと「TYRL.SUF」との位置関係が固定で，画面上での動きがないのも不自然です。カメラを追い越すように，少しパンをかけたほうがよいでしょう。

さらに，カメラの位置ももう少しアップ気味に，かつ下に設定したほうが迫力があります。「TYRL.SUF」も等速運動では味気ないので，後半加速させます。このために，1フレーム目と，50フレーム目だけでなく，30フレーム目も設定します。

○相対物をつける

概要：車が走るシーンの相対物としては，センターライン，ガードレール，トンネルなどもありますが，背景が野山なので，木を植えることにしましょう。しかし，リアルな木をCADで作るのは非常に難しいので，今回は回転体か，角柱と角錐を組み合わせたような簡単な木でごまかすことにしましょう（誰か木や草を作って，ディスクマガジンに投稿してくれ〜）。

CADの使い方についてはここでは省略させていただきます。マニュアルの「CAD入門」や図8を参考にがんばってください。物体名は「TREE」にします。また，面数が多いと作画が大変なので，あまり増やさないように。それから，「TREE」のアトリビュートファイルもちゃんと作ってください。

操作：1) 「X11A.FSC」を読み込む
・FFEを起動する
・「ファイル」の「LOAD」で「X11A.FSC」を選択する
・「＝」キーを2回押して，表示範囲を変える

2) 木を植える
・「物体設定」の「追加」で「TREE.SUF」を選択する
・図9の範囲をマウスでクリックする
・X座標が−300以下であることを確認する
(−300以上の場合は別の場所をクリックする)
・「決定」
・再び「追加」をし，「TREE.SUF」を植えていく
・十分な数に達したら，「物体設定」の「終了」でメインメニューに戻る

X座標の−300〜300は「TYRL.SUF」が通る幅です。また，300以上は，カメラの後ろになりますので，木を植えても見えません。木は，−400〜−300の範囲，つまり，「TYRL.SUF」のすれすれの位置にたくさんあるほうがスピード感が出ます。20〜30本も植えればよいでしょう。

○1フレーム目の修正

操作：視点の修正
・「視点設定」で視点の座標を(200, 3200, 90)にする
・注目点の座標を(0, 3400, 70)として「決定」

○50フレーム目の修正

操作：1) フレームナンバーの変更
・「フレームNo.設定」でフレームナンバー「50」を入力して，リターンで決定する

2) 視点の修正
・「視点設定」で視点の座標を(250, −2400, 90)にする
・注目点の座標を(0, −2700, 70)として「決定」

○30フレーム目の設定

操作：1) フレームナンバーの変更
・「フレームNo.設定」でフレームナンバー「30」

図8 TREE.SUF（Z座標：0〜300程度）

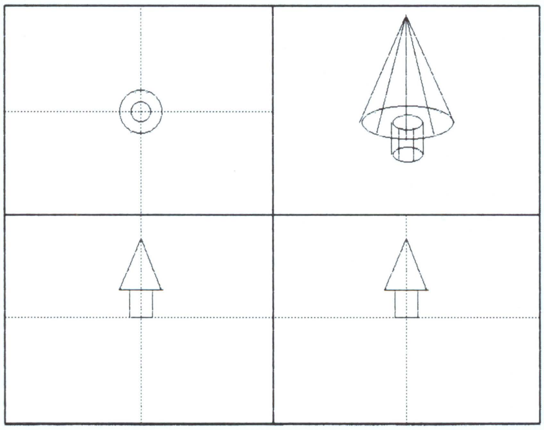
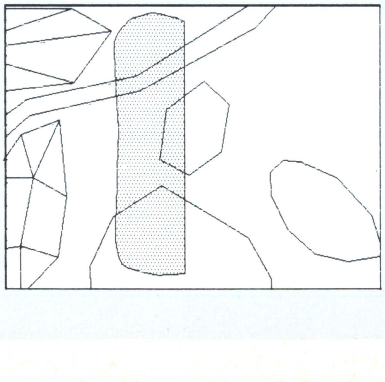


図9 木を植える範囲



を入力して，リターンで決定する

2) 「TYRL.SUF」の変更

・「物体設定」の「変更」で「選択」を何回かクリックして，「TYRL.SUF」を選択

・位置座標はYだけ変わって(0, 0, 0)で「決定」

3) 視点の変更

・「視点設定」で視点を(250, −200, 90)，注目点を(0, −200, 70)として「決定」

4) セーブ終了

・「ファイル」の「SAVE」で「フレームソース」を選択し，出力ファイル名「X11B」をキー入力する

・「終了」でFFEを終わる

○作画

操作：1) WIREVIEWでの確認

・「FF X11B」を実行

・「WIREVIEW TYRL.SUF JIMEN.SUF TREE.SUF X11B.FRM /V」を実行

・異常がないか確認し，「ESC」で終了

2) 作画・アニメーション

・「AUTO X11B.FSC-A2 -G-D」を実行

・飽きたら，「ESC」で終了

3) 比較アニメーション

・「MKTCH X11A001 X11B001」を実行して，連続したタイムチャートを作る

・「HANIM X11A」でアニメーションを再生

いかがですか？ 2つのカットを見比べてみるとスピード感にずいぶんと差があるのがおわかりいただけたでしょう。

フラッシュカット

もうひとつ簡単なカットを作ってみます。20フレーム，つまり1秒という短いカットです。カメラの位置を極端に低くして「TYRL.SUF」に寄せることで，スピード感を強調します。

操作方法はもう十分修得されたでしょうから，データを紹介するだけにとどめます。

1) 1フレーム目の設定

・背景を同様に設定

・「TYRL.SUF」を(−1000, 0, 0)でZ軸回りに5度回転

・視点を(1000, −250, −55)，注目点を(−800, 100, 500)

2) 10フレーム目の設定

・「TYRL.SUF」を(600, −250, 0)，角度を−15度に変更

3) 20フレーム目の設定

・「TYRL.SUF」を(1200, −700, 0)，角度を−30度に変更

・「X11C.FSC」でSAVE，終了

4) 作画・アニメーション

・「FF X11C」を実行

・「WIREVIEW TYRL.SUF JIMEN.SUF X11C.FRM /V」を実行

・「AUTO X11C.FSC-A2 -G-D」を実行

編集

「X11A」はボツにするとして，「X11B」と「X11C」を前回のアニメーションとつなげてみましょう。

1) そのままつなげる

まず，とにかく全部を順番につなげてみます。映画でいうところのラッシュフィルムです。前回作った「X10A.TCH」にエディタで2カットつけ加えます。

```
.timechart
.wait 20        x10a001
.wait 1         x10a [2-19]
.wait 10        x10a020
.wait 1         x10b [1-5]
                x10c [1-40]
                x11b [1-50]
                x11c [1-20]
.endchart
```

ところで，フロッピーディスク上で作業されている方は，全部の画像データが1枚に収まりきらないでしょう。その場合は「X11B」と「X11C」を別のフロッピーディスクに入れることになります。

たとえば，Aドライブがハードディスクで，Bドライブに「X10A」「X10B」「X10C」と「X10A.TCH」があり，Cドライブに「X11B」「X11C」がある状態で，「X10A.TCH」をアニメーションさせると，途中で，「can't open 'x11b001.pic'……」と表示されます。

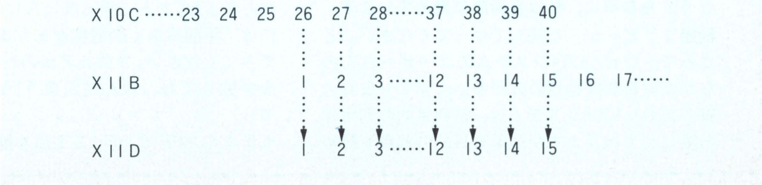
このようなときは，ディスクを入れ替えてもいいのですが，「C」のキーを押せばCドライブにファイルを探しにいきます。

また，最初から「X10A.TCH」の最後の3行を，

```
        C:x11b [1-50]
        C:x11c [1-20]
.endchart
```

としておいてもよいでしょう。

図10 オーバーラップ



オーバーラップをかける

アニメーションをご覧になっていかがですか？　この編集には問題点が2つあります。ひとつは「X10C」と「X11B」とのつなぎがやや不自然なこと，もうひとつは，「X11C」が若干間延びしていることです。

まず，ひとつめの問題は，オーバーラップによって解決してみましょう。「X10C」の最後と「X11B」の最初のそれぞれ15フレームを「OVLAP」というツールで合成した画像を作り，「X11D」という名前でSAVEします（図10)。そしてコマンドラインから，

OVLAP X10C026 X11B001 /F15 /OX11D /D

を実行してください。

「OVLAP」は，2つの画像ファイルを指定しますが，先に指定した画像から，あとに指定した画像にだんだん変わっていく画像を作ります。「/F」オプションは，オーバーラップさせるフレーム数で，数が多いとゆっくり変化します。

「/D」は動画指定オプションです。これがないと「X10C026.PIC」「X11B001.PIC」という2枚の静止画から，割合を変えながらその中間的な画像を作り，「X10C027.PIC」〜「X10C040.PIC」や「X11B002.PIC」〜「X11B015.PIC」を使用しません。意味がわからなければ，一度やってみて，見比べるのもよいでしょう。

なお，オーバーラップをすると，一般に色数が増えますので，CRDで色数を落とす必要があります。「CRD X11D /OX11D」を実行してください。

再編集

2つめの問題点だった「X11C」の間延びについては，タイムチャートを書き換えて，前後数フレームを削ります。また，オーバーラップした画像も入れ，その分「X10C」と「X11B」を15フレームずつ削ります。

```
.timechart
.wait 20      x10a001
.wait  1      x10a [2-19]
.wait 10      x10a020
.wait  1      x10b [1-5]
```

読者によるほっとけないほっとこらむ

今回は，うさ子さんがお休みなので，代わりに姫がお便りを紹介します。ちょっと緊張，へへっ。うさ子さんはディズニーランドに行きました。いいな，姫も行きたいなあ（関係ないけど，ドナルドのおしりって，かわいいね)。

ところで一昨日，大学で研究用の九官鳥（何に使うんだろ？）のカゴの掃除をしているとき，ふと見ると暗幕の下に黒のレースがついている。こんなところにレースなんてつけて，まったく先生もおちゃめなんだから……と，ぶつぶついいながらカーテンを引っ張り上げました。すると，あーら不思議，レースはほどけてかさかさと走り出しました。

ぎゃー!!!!　ご，ごきぶりだー!!!!

あとの惨状はご想像におまかせします。姫は黒のカーテンが嫌いになりました。

でも，鳥さんが手からえさを食べてくれると，とってもうれしくなります。あいさつもしてくれるし。でも，入ってきて，いきなり“バイバイ”といわれると，ちょっと悲しい。

＜Aさん＞現在高3生。京大か阪大に合格したらDōGAに入りたいです。合格発表のときに勧誘しているスタッフの方を発見できたら，そちらに走っていきますので，堂上げよろしくお願いします。

柚姫：ぜひぜひ来てくださいね。だけど，堂上げじゃなくって，胴上げじゃないの？　こんなんで大丈夫かなあ，心配。

＜Bさん＞DōGAはなんと読んだらいいのでしょう。ちなみに，私は自衛隊に勤めています。

柚姫：「どーが」と読んでやってください。ところで，なぜかCGAシステムのユーザーさんのなかには自衛隊勤務の方が多いんです（日立の寮の方も)。CGAシステムは，営利目的の利用はお断りしておりますが，軍事目的の利用も認めませんよ。ちゃんと平和利用してくださいね。

＜Cさん＞CADについてなのですが，オブジェクト制作中，約619面にくると突然，各命令が反応を示さなくなります。これはプログラム上の面の上限の設定に到達したためなのでしょうか。

柚姫：メモリの大きさによるのですが，2Mバイトで全部4角形だった場合，少なくとも4000面以上いけるはずなんだけどなあ。上限は面の数ではなく，点の数だから，30角形でも作ってました？　いちおう，今度配布したバージョンでは上限を変更することもできます。詳しくはマニュアルを読んでくださいね。

ちょっと原因がわからないので，よろしければそのデータを送ってください。それとオブジェクトを作るときには，各部品に分けて作って，最後にKAMA.Xでくっつけるほうがいいですよ。

＜Dさん＞3点めの点固定で暴走するため，CADでオリジナルの形状が作れません。

柚姫：やっぱりなんのことだかわかりません。トラブルについては，できるだけ詳しく書いてください。こちらで再現しないことには，対応しようがありません。

まず「点固定」は「点確定」のことですよね。暴走ではないですが，3点が同一直線上だと，「点確定」はできません。さらに，3点めではありませんが，4点めで「点確定」できないことはあります。その4点めが，すでに確定している3点によって作られた面（高校の頃習いましたね。3点が決まると，平面がひとつ決まるんだったよね）の上に乗っていないためです。CADには「平面投射」の機能がありますので，使ってみてください。マニュアルのT-308（CAD補講）を勉強してね（ぜんぜん違う問題だったりして)。

＜Eさん＞FFで，div1で3点を補間させると変な値が出るバグは小さな問題ではありません。早く直してください。

柚姫：FFの作者の山下さんはいらっしゃらないんですけど，どなたかわかりますか。あっ，恒光さんお願いします。

恒光：ソースを見てみます。ふむふむ，あっ確かにおかしいですね。でもすぐ直りますよ。ちょいちょい。

柚姫：ということで無事バグは取れました（こんなに簡単に取れるなら，最初から取れって？)。ついでに，高津さんが文月さんの強引な直訴により乱数関数を加えました。このように，CGAシステムは，日夜バージョンアップを続けています。FFEなんかもとっても使いやすくなりましたよ。

かまた：これらのバージョンアップツールは年内にまとめて，TAKERUなどで発表する予定です。ディスクマガジンの準備もせんとあかんし，ああ忙しい。

＜Fさん＞質問をまとめて書きます。

1)　Z'sTRIPHONYの物体データをコンバートすることはできませんか。

2)　PC-9801用のCGソフト「3D-STUDIO」（価格は800,000円）もCGAシステムと同じスキャンラインだそうですが，鏡の映り込みや影を表現できます。これはアルゴリズムの差ですか？値段の差ですか？

3)　トランスピュータを手に入れましたが，RENDをトランスピュータに対応できませんか？

柚姫：なんか難しい質問ばっかしだな〜。え〜い，かまたさん，まかせた！

かまた：まかされてしまった……。

1)　Z'sTRIPHONYのみならず，PC-9801用のCADなど，いろんなコンバータがネットで出回っているそうです。しかしながら，作者の方と連絡

x10c [1-25]
x11d [1-15]
x11b [16-50]
x11c [4-17]

.endchart

いかがでしょう。だいぶよくなったと思いませんか？

おわりに

今回いいたかったことは，“センス”とかいう天から与えられた魔力と思われていることの大部分が，知っていれば誰にでもできるテクニックでしかすぎないということです。

これで，FFEによるモーションデザインの方法と，よくひっかかる問題点の解決法を解説しましたので，打倒TORNADOもあとは実行に移すだけ。この続きのカットは自分でどんどん制作して作品にまとめてください。

しかし，本当にこれだけでTORNADOと同等の作品ができるのか？　何か足りないぞ？　何か忘れているぞ？

最初の予定では前後編だった「打倒TORNADOへの第一歩」を長々と続けておりますが，いよいよ次回は最終編です（連載が終わるわけではありませんよ)。“本当に誰にでもTORNADOはできるのか？”という謎(？)がついに解明されます。お楽しみに。

当チームおよび，この連載・各コラムに関するお問い合わせなどは，下記の住所までお願いいたします。

〒533　大阪市東淀川区淡路5-17-2 102号
プロジェクトチームDōGA「あてにならないアフターサービス」係

その他のコラム

法人化への道
遅々として，進まず。

CGAマガジン編集部より
データ集まらず。発行危うし。

面倒くさいCGAシステム配布係より
９月号の実費2,000円は3,000円の誤植です。ディスクがつくと，書籍小包では送れないので郵パックになるためです。うっ，赤字……。

CGAコンテスト事務局より
作品制作がんばれ！　今年は，４カット部門が穴場と見た！

がとれないため，当チームから配布するわけにはいきません。バージョンアップのとき，一緒に入れたいところですが……。作者の方がこれを読んでましたらご連絡ください。ちなみに，TOYBOXからのコンバータはGRさんが責任を持って発表してくださるそうです（誌面上で催促したりして)。

2)　たしかにスキャンラインでも，映り込みや影を表現することは可能です。しかし，表現力を上げれば上げるほど，作画スピードが落ちます。CGAシステムではアニメーションが目的ですので，速度を優先したレンダリングを行っているわけです。速度を上げることも，表現力を上げることも，ともに難しいわけです。REND.Xのほうが安物だから機能が落ちるというわけではありません。

3)　トランスピュータに対応するためには，当方がそのトランスピュータを持っていないといけません。また，多くの方がそのトランスピュータを持っていないと，移植する意味がありません。ということで，技術的には可能でも，あまり積極的に行う予定はありません。

トランスピュータの話が出たところで思い出しましたが，前号予告した作画スピードのベンチマークテストの発表です（表１)。

作画条件によって値が変わるのは，浮動小数点計算が占める割合が違うためです。また，X68000の数値演算ドライバはコプロなしの場合は「FLOAT2」を，コプロがある場合は「FLOAT3F」を使用しています。RENDXVIはコプロに直接アクセスするため，ドライバには依存しません。

PC-486GFの24MHzというのは改造品で，GR相当と思っていただければ結構です。それにしても，52.89倍というのは驚異ですね。

でも，V70ボードが24MHzに改造したXVIに負けるのはショックでした。これは，V70の性能を生かすソフト的な環境が整っていないためで，決してV70が悪いわけではありません。単にCPUを速くしても，そのマシンを生かしたソフトを走らせないとまったく意味がないことは，ほかのマシンのデータからもわかります。

さらにショックなのは，LUNAのデータを見てみると68030でも，速度的にはあまり速くならないことです。コプロが68881であることや，システムなど，さまざまな要因が影響しての数字なのでしょうが……。

表１

●作画条件　ポリゴン数：7600面　アンチエイリアス２倍　スムースシェーディング

マシン	CPU(周波数)	コプロ	プログラム名	時間	速度比
X68000ACE	68000(10)	なし	rend	16m34s77	1.00
X68000ACE	68000(10)	68881	rend	9m55s46	1.67
X68000ACE	68000(10)	68881	rendxvi	5m33s49	2.98
X68000XVI	68000(16)	68881	rend	7m07s66	2.33
X68000XVI	68000(16)	68881	rendxvi	3m08s32	5.28
X68000XVI	68000(24)	68881	rend	4m02s05	4.11
X68000XVI	68000(24)	68881	rendxvi	2m01s32	8.20
X68000ACE	V70(20)		rendv70	2m11s31	7.60
PC-286VF	80286(12)	2C87	rend98	18m48s75	0.88
PC-286VS	80286(16)	なし	rend98	23m50s92	0.70
PC-486GF	80486(16)	ODP	rend98	3m10s86	5.21
PC-486GF	80486(16)	ODP	rend386	0m30s27	32.86
PC-486GF	80486(24)	ODP	rend386	0m18s81	52.89
LUNA	68030(20)	68881	rend	1m53s37	8.77

●作画条件　ポリゴン数:150面　アンチエイリアスなし　スムースシェーディングなし　５フレーム

マシン	CPU(周波数)	コプロ	プログラム名	時間	速度比
X68000ACE	68000(10)	なし	rend	1m41s17	1.00
X68000ACE	68000(10)	68881	rend	1m18s48	1.29
X68000ACE	68000(10)	68881	rendxvi	1m00s22	1.68
X68000XVI	68000(16)	68881	rendxvi	0m36s72	2.76
X68000XVI	68000(24)	68881	rendxvi	0m26s32	3.84
PC-286VF	80286(12)	2C87	rend98	1m01s28	1.65
PC-286VS	80286(16)	なし	rend98	1m27s73	1.15
PC-486GF	80486(24)	ODP	rend386	0m05s13	19.72

TORNADO発進準備完了

文月 涼

本当は“芸術祭版TORNADO”のビデオが発売になってから，それを題材に皆さんに話をしようと思っていました。が，そのビデオの発売いかんは，先行して発売される芸術祭出品のゲーム集の売れ行きにかかっているそうなので，先に連載を開始して前人気を煽ろうと突然思い立ちました。

7月号掲載の拙著「4DCGへの招待」では，主に心構えみたいなものを紹介しましたし，DōGAの連載では初心者向きの講座が開講されていますので，このコラムではTORNADOを題材にとって，私が身につけたハイテク（姑息な技）の数々について解説をしていきたいと思います。

ver.2.50を機にCGAシステムを使い始めた人には，いきなり難しい話になるかもしれません。その場合はリアルタイムに理解しようとはせず，自分のペースでCGが作成できるようになってから読み返して参考にするといいでしょう。将来的には，まとまったかたちで皆さんにお渡しできたらいいなとは思いますが，そこはそれ，なにですから。

予定ではコラムは，

○基本方針発表（今回）
○トータルイメージの設計
○モデリング
○カラーリング
○目の錯覚
○光と影
○構造体
○画像合成
○映像製作のまとめ
○デジタルメディアとの同期
○著作権を取得する
○ビデオの作成

などをポイントにして，進行しようと考えています。いずれビデオが発売になったら，シーンごとのケーススタディも行いたいと思っています。

また原則として，DōGAの連載と重複するような初歩的事項には触れません。より多くの情報を皆さんに伝えるために，できるだけ質実剛健でいきたいと思います。

トータルイメージについて

では，手始めにトータルイメージについての考察を行いたいと思います。

CGAを芸術としてとらえた場合，それは単純にほかの芸術と比較すれば高次元の芸術であると考えます。これは決して各芸術の優劣を位置づけているのではなく，ただ単に1次元，2次元，n次元と次元が並んでいるという意味です。

CGAは映画にも似て，絵画，物語，音楽などの単一メディアとは異なり，これらのものを合体した分だけ高次元にあると考えられるのです。単一メディアでは想像力が要求され，また，そのことが作品を昇華させます。しかし，人間をとりまく自然の環境の中では，それぞれが個々で存在することはほとんどなく，スケジュールと音と視覚が常に複雑に絡み合って人生をかたちづくっています。映画やCGAはその，より自然な人間の完成に近づいていくメディアであるという意味なのです。

したがって，それぞれの同期，調和という点において，個々の芸術を遂行する場合よりも，より完成された展望が要求されるのは事実です。当初からなんらかのトータルイメージをもっていることは，むしろ必要不可欠であるといえるでしょう。

あなたはなぜCGAに興味があるのでしょうか。プレゼンテーション，映像制作業，あるいは趣味？ CGAそのものを作りたかったり，アニパロという線もありますね。要はなんでもいいんですが，作ろうとするものが完成した姿を一度，頭の中に思い描いてみてほしいのです。

いきなりそういわれても思い浮かばない場合は，自分が好きな曲や1枚のイメージスケッチをもとにして想像をふくらませるのもよいでしょう。注意すべきは，これから作るのは動くモノですから，イメージが流体であってほしいのです。でなければ，レイトレーシングを使った1枚のCGでもかまわなくなってしまいます。動く，ということを忘れずに。

動く，については難しく考える必要はありません。あなたはふだんから動いて生活しているのですから，呼吸するのと同じように動く物体を描いていくだけなのです。

作品のイメージが完成したら，そのイメージをもとに，作業はモデリングへと進みます。

モデリングは自分が作ろうとする作品の完成像によって異なってくるのですが，最初から凝ったものを作る必要はまったくありません。CGの利点はほかの芸術と異なり，やり直しに必要とされる人的労力が極端に少ないのです。初期の段階で簡易モデルを使って，あとで形状モデルを複雑なものに差し替えたとしても，必要とされるのは，大部分がマシンパワーなのです。

たとえばTORNADOでは，初期に作成した形状モデルを現在まで16回バージョンアップをしています。しかし，レンダリング用のフレームファイル（動きの指定をしたファイル）に関しては，ほとんどいじっていません。つまり，フレームファイルを一度作成してしまえば，何度形状モデルを差し替えても，ロスするのはマシンをこき使うレンダリングの時間だけなのです。しかし，これがモデリングの次に時間がかかる作業でもあります。ただしこの状態では作った本人はまな板の上の鯉なのですが。

だから，私はいいきってしまいます。
“モデリングにいきなり凝るな！”
“そんな時間があったらまず完成を見よ！”

作品の全貌が固まらないうちの形状モデルのイメージは，あいまいなままでいいでしょう。最初は簡単な形状モデルを使い，不満があればあとでモデリングに凝ればいいのです。CGAシステムを使おうとしたユーザーの多くがモデリングで挫折しているのですから。

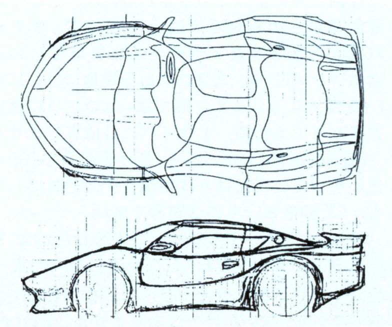
以下は，そのことを念頭においての「モデリングについて」です。

モデリングは対象とする形状を把握するところから始まります。当然自分が作ろうとする映像の完成像とうまく溶けあうかたちである必要があります。個々のモデリングがかっこよくても，作品全体を見渡すと，そのモデルが浮いてしまっていることはままあることだからです。

立体的な感覚が優れている人は，いきなりCADに向かってモデリングを開始してもいいのですが，普通の人はまず対象とする物体を方眼紙などに落とし，3次元のイメージをつかんだほうが無難でしょう。

作品の全体像から浮き出ないイメージスケッチをもとに，横，上，前からまずラフなイメージを描きます。いきなり整然とした図面を描く必要はありませんし，各図面間での整合性をとる必要もありません。まずはくだらないことに気をとられないで，自分の感性のおもむくままに，イメージを紙に叩きつければいいのです（下図参照）。思いのたけをぶつけたら，その状態で図面を一度コピーしておいて，清書に移ります。ラフスケッチの上からモデリングすることを考えながら，もう一度フォルムのラインを描き込んでいくのです。

では，モデリングしやすいラインとはどのようなラインなのでしょう。それはまた次に。



[特集]

ゲームマネージメント

ある種のゲームはパーソナルコンピュータの性能をもっとも発揮するアプリケーションである。画面上には趣向を凝らした華麗な世界が次々に展開される。
しかし，その背景は決して華々しいものではない。芸術的なテクニックはむしろ泥沼に近いコーディングから生まれてくるものであろう。クロックを数え，アルゴリズムを磨き，見事に最適化されたプログラムは一般のプログラムとは一線を画した，きわめて特殊な世界を形成する。その1つひとつの驚異を構成しているのが特殊化されたサブルーチンであり，それぞれのマネージャであるといえる。
見るものを驚かすハイテクニックの数々。それも，いつしか当たり前の要素として取り込まれ，さらに発展していく。こういった技術の蓄積は作品の水準をいやおうなく上げていくだろう。蓄積は力である。
逆にいえば，そういったプロフェッショナルな仕事を期待できる部分が揃っているならば，ゲームの本体の負担はぐっと軽くなる。本質に力点を移すことができるようになるだろう。
無論，ゲームの本質は小手先のルーチンにあるのではないのだ。

CONTENTS

概論
ゲームシステムの構成……………………中野　修一

よりよいゲームを作るために
違いのわかるアソビロジー…………伊涵見　あきら

リアルタイムゲームのための
"基本"スプライトルーチンの作り方………横内　威至

ゲームマネージャの作成
BEMSによるキャラクタ管理……………影山　裕昭

考え方と実例を探って
オブジェクト指向を取り入れる………………丹　明彦

シナリオ管理の手法
アドベンチャーシステムを考える…………香坂　正嗣

プログラム速度制御関数
XVI.FNC……………………………………紙山　満

BASICのスプライト処理高速化
SPRITE 2 .FNC………………………………中村　文洋

グラフィック画面の3D回転
3DRT_256……………………………………渡辺　貴生

イラスト協力：高橋　哲史

概論

ゲームシステムの構成

Nakano Shuichi 中野 修一

ここではプログラミングスタイルとしてのゲームマネージメントを考えてみます。どのような考え方でマネージャを作成するのか，マネージャとデータの関係などを見てみましょう。

プログラムを作成するうえでは，誰でもその人なりの作法なり考え方というものがあると思います。本来，プログラムという行為自体には色がないというか，なにをしてもかまわない自由さがあります。しかし，人は自分色を持ちたがるもので，プログラムにもいろいろな流派があります。

速度優先，オブジェクト指向，モジュール化，遺伝的などの言葉を挙げるまでもなく，さまざまなパラダイムが連立している世の中ですが，ゲームプログラミングに関してみると幅はかなり狭いようです。

ゲームにおけるひとつの典型は，昔のBASICプログラムでよく見られたように，メインループ以外の構造がほとんどないものです。必要になったらその場で必要なだけの処理ルーチンを加えていきます。場当たり的といえば，確かにそうですが，完成時にもっとも高い性能を示す（可能性がある）のはこのタイプです。

なにかのゲームを見て感銘し，似たようなものを作りたいと思う人がいたとしましょう。最初はすべてまとめてプログラムするのもよいでしょう。しかし，おそらく，いくつかのゲームを作るうちにいちいち細かいことを指定しなくても，ある程度のことを代わりにやってくれるものがほしくなります。

たとえば，スクロールシューティングゲームでは，それがどんなものであれ，なんらかのかたちでマップデータをメモリ上に持つことは間違いありません。そして，その仕様は，たとえ10人が別のものを作ったとしても概要に大差はないものになるでしょう。であれば，適切なデータ形式と多少柔軟な構造を持ったルーチンを作成して使い回しをすることを考えても不思議はありません。

いくつかのゲームを取り出して，それを構成する要素を分解していくと，それぞれ，なんらかのデータを管理するプログラムとデータ自体に分けることができます（以後はマネージャ，データと表記）。

データはプログラムの中に埋め込まれていることも少なくありませんが，うまく処理すれば分離することも不可能ではありません。

それらのマネージャ自体も下位ルーチンを利用していることが往々にしてあり，また，それぞれの下位ルーチンは限定された意味でマネージャの一種であるともいえます。

つまり，プログラミングの流れを，

下位ルーチンからマネージャを作る

データを用意しマネージャを使う

のようにすることができます。小さなものを集めて大きなものを作るボトムアップ的なアプローチですね。突き詰めていえば，マネージャとデータを揃えてやればゲームはできあがる，ということです。

これが今回のお題です。

念のため騙されやすい読者のためにフォローしておくと，これは「ゲームが作りたい→プログラムを作ろう」というのを「マネージャとデータを作ろう」にいい換えただけで，本質的には進歩した点はなにもありません。しかし，これが「アルゴリズムとデータ構造を作ろう」でないのはなぜかというと，問題の捉え方を変えること自体に意味があるからなのです。

ゲームの宇宙

基本ルーチンを用意してプログラム作成効率を上げるという基本的な考え方は，X68000のDOSコールやIOCSコールの生まれた背景と特に変わりはありません。

ですが「積極的にIOCSコールを使いましょう」という展開にならないのは，題材を「ゲーム」と限定しているからです。統合された環境が喜ばれるなかで独り遊離した位置を保っているのがゲームプログラムです。内容がよいものであれば，違うOSで動いていてもあまり文句をいう人はいないでしょう。別の宇宙を作ることが認められた唯一の分野かもしれません。

逆にいえば，それだけたくさんのものを新しく作り直さねばならないということにもなります。実際，あらゆるマネージャが求められています。

ここでマネージャといっても，それにはさまざまなレベルがあります。それは単純な文字表示かもしれませんし，3次元処理かもしれませんし，ほんの小さなノウハウかもしれません。

なにが必要か?

そこにどのような宇宙が展開されるべきなのでしょうか。

リアルタイムゲームではグラフィックやスプライト，テキストなどの各ハードウェア資源を効率よく適切に運用することが必要となります。すなわちハードウェアに密着した高性能ルーチン群が要求されます。

ここでは，エラー処理を大胆に省略しても許されますし，メモリ効率などを無視したコーディングも容認されます。まず，性能が問題。こうなるとIOCSでは役不足，流行のインライン展開とテーブル化が花開きます。場合によったらOSから用意するのが正しい姿なのかもしれません。なにしろアーケードゲームでは特定のゲームのためにハードウェアを開発することもあるのですから。

スプライトを持たない機種ではソフトウェアでスプライトを実現するという手もあります。ソフトウェアを介在させてはスプライトのメリットがないという方もいるでしょうが，同じ入力で同じ出力結果が得られれば過程はあまり重要ではありません。特殊な例ですが，セガではスプライトという言葉を「キャラクタ」の代わりに使用し

ていますので，画面表示関係ならなんでもかんでもスプライト表示システムとなります。同様に考えればソフトウェアスプライトというものもあっていいでしょう。

次に，特殊機能です。MAGICのような３Dカーネル，さらにポリゴンエンジン，グラフィックやスプライト回転システム，今回の3DRT_256のような３D表示システムやラスタースクロールシステム。音楽でいえばZ-MUSICクラスの演奏システムなどでしょうか。

そしてさらに高機能なキャラクタ管理システムとしてキャラクタ単位の移動やアニメーション，衝突判定などをまとめてやってくれるものがほしくなります。

アドベンチャーゲームなら基本表示システムとシナリオインタプリタを作成しておけばかなりのことはこなせます。

RPGはアクション系か否かで基本システムが変わりますがパラメータ評価システムとフラグ管理システムが中心となるのは間違いないでしょう。もちろん，アドベンチャーゲームとの複合型もありえます。

シミュレーションゲームの場合，種類が多すぎて難しいのですが，ひと言でいえば表示システムと評価関数です。ただし，汎用の評価関数というものを用意するのは難しそうですけど。

＊　　＊　　＊

これらのうちいくつかはすでに実現されています。MAGICでは，決められた書式に従ってデータを用意しておけば，あとは簡単なコマンドを送るだけで3Dワイヤーフレーム画像を表示します。

CARDDRVも同じように番号と表示位置を指定するだけで手軽に美しい画面表示を実現します。このドライバがなかったらカードゲームの作成はかなり困難なものになるだろうことは想像に難くありません。

最近のゲームでもっとも大きな比重を占めているのは表示部分です。MAGICもCARDDRVも表示部分だけで大きな効果を上げています。

さらに表示にこだわってくると必要になるのはアニメーション管理でしょう。パターンが少ないうちは個別に処理しても大差ありませんが，すべてのキャラクタを滑らかに変化させようとするとデータの管理だけでも大変です。

たとえば，アニメーションデータに従ってスプライトを自動的に書き換えるマネージャが作成できればスプライトの最大定義数の壁を突破することもできそうです。

それでも垂直帰線期間内に書き換えが可能なPCG個数には限界がありますから，キャラクタごとに待ち行列に並べて1/60秒ごとに時分割処理するような工夫は必要でしょうし，極度に大きなアニメーションパターンは作成できないといった制限は出てくるでしょう。

ハードウェアの補完

高度なマネージャは，ときとしてハードウェアを拡張したかたちで扱うことができます。もともとソフトウェアとハードウェアは同じ目的を持ったものであり，表現方法が異なるだけなのです。時間的にクリティカルな部分が発生する以外は，ハードウェアでできることならソフトウェアでも同じことができます。

スプライトの拡張であれ，PCMの変調であれ，グラフィックの回転であれ，ソフトウェアで処理することが可能です。

このようなかたちでマネージャを拡張し，システム環境を整備していくことができれば，プログラマの負担はどんどん軽くなり，表現力はどんどん上がっていきます。

ツールの充実

マネージャができたらそれに与えるデータが必要です。

マネージャとしての独立度が高まるほどデータは重要な役割を果たします。たとえ優れた技術によるマネージャが用意されていても，吟味されたデータを供給する手段てがなければ十分に活用できません。

現状ではゲームを作る場合，それに先立って開発ツール一式を自作する必要があります。なんだかおかしな話です。

マネージャとデータという分類では，プログラムの仕事とデータの仕事を分離してやるということが基本になっています。また，システムの性能を追求するとデータ形式というのは往々にして一般性を欠いたものになりがちです。ですから，データを分離するという作業は，そのデータを作成できる環境を揃えてからでないとまったく意味がありません。

データを分離する際には既存のデータをいかにしてそこに流用するか，あるいはどのようなエディタを作るかということも念頭に置かねばなりません。

ゲームだからこそ

ゲームでは効率を追求することが第一で，必要ないと思われることはいっさいしないほうが美徳であるともいえます。実際，エラーチェックなども最低限しかしない場合がほとんどです。そんなソフトウェアを走らせて危なくないのでしょうか？

しかし，実際のところ，ユーティリティや実務ソフトよりもゲームソフトのほうがバグ発生率が遥かに少ないのではないかと指摘する人もいます（「ク」で始まる接頭語を戴かないゲームに限っていえば）。

信頼性が二の次なのに危険でないというよりは，信頼性を第一にしていながら不安定なものが多すぎるからでしょうか。時速100kmで障害物にぶつかった場合，安全性を考慮されているはずの自家用車よりもF1マシンのほうが事故時のダメージが少なそうな気がするのと似ています（試したことはないけど）。

一般的なソフトのほうが複雑だからという考え方もありますが，ゲームというのは割り込み処理やリアルタイム制御の山なので，適切な指摘とはいえないでしょう。

遊びのためのプログラムだからこそ，もっとも洗練されたプログラミングテクニックが要求され，感性に心地よいLOOK&FEELを持ったユーザーインタフェイスが求められる，ともいえるでしょう。

システムを磨け

ということで，今回は「ゲームプログラミング」という混沌とした塊からいくつかの方法論を切り出しています。

ずらっと並んだ記事項目を見ても，実にさまざまなものがあります。一部重複するものがあったり，矛盾する部分もあるかと思います。はっきりいえるのは，それぞれは，ひとつのアプローチにすぎず，いうならばまだまだ磨かれていないということです。こういった類のプログラムには，いくつかの実践的なプログラムに饗して初めてわかる問題点も多くあります。MAGICやZ-MUSICなどはそうして磨かれてきたのです。

そして，正解といえるものも存在しないということです。ひとつの方向から問題を切り出した時点で，あらゆる視点からの検証に耐えられるものであることは放棄されたといってもいいでしょう。

では，これらのアプローチからなにを生み出せるのでしょうか。それは，これらを磨き上げていく皆さんにかかっている問題です。これらを生かす活用法を見つけ出してみてください。

よりよいゲームを作るために

違いのわかるアソビロジー

Ishibumi Akira 伊溢見 あきら

なまじマネージャなどができると，よく似たゲームというものも多く現れてきます。同じシステムを持つもののなかでも，面白いものとそうでないものが厳然と存在します。いったい，なにが面白さを決めるのでしょうか？

誰でもコンピュータゲーム(以後ゲームと略記)を遊んだことはあると思います(Oh!Xの読者なら間違いないでしょう)。ゲームとは自己の作用に対して反応が返ってくるものですから，それに対しての印象や感想を持つことは，非常に自然な行為といえます。そういったものは，そのゲーム固有のイメージとして，プレイヤーに刻み込まれていき，それらの経験を比較することで「面白い」「つまらない」といった分類ができることは，どんな人でも経験していることだと思います。

しかし，なぜそういったイメージにたどり着くかは，あまり明らかにされていません。そこで，今回の特集の「ゲームを作る」という目的を考えに入れ，普段のプレイでは見えない部分を，少しだけでもわかりやすく説明してみたいと思います。できるだけ，気をつけて書いていくつもりですが，難しい場所やわかりにくい部分があった場合はお許しください。では，気ままに遊ぶだけに見えるゲームに，どれだけの理論が隠されているのかを探っていきましょう。

類似品にご注意

ゲームを作ろうという目的を持ったときには，当然誰もが面白いゲームを作りたいと思うはずです。手っとり早い方法として有名なのは，すでに面白いという評価の定まったゲームを模倣することでしょう。

ほかにも，異なった環境にある面白いゲームを手近な環境(たとえばX68000)に持ってくるという移植の場合も，これとほぼ同じと考えられます。実際に世の中にそういったゲームは数多く，最近はどれがオリジナルだかわからなくなっているものまであるようです。画面構成やキャラクター設定に始まり，簡単な演出までソックリにしたものも珍しくありません。ちょっとゲームに詳しい人に「1対1の対戦格闘ゲーム」という条件に当てはまるゲームタイトルをわざわざ挙げてもらうまでもなく，類似ゲームの氾濫は，昔から続いてきた普通の状況だということがわかります。

しかし，それらがすべてオリジナルのように面白いかというと，迷うことなく誰もが否定するに違いありません。ではなぜ面白いものを真似しているにもかかわらず，面白くないものができあがってしまうのでしょうか？。

それは，そのイメージが生まれる過程を考えてみることで解決します。最初にも書いたとおり，面白いというのは遊んだことで与えられたゲームの反応に対してプレイヤーが判断した結果です。この結果はあらゆるゲームに対して，直接にはなんら具体的な制約を与えませんし，あくまで個人の判断ですから絶対的なものでもありません。ということは外見や形態を模倣したとしても，面白いというイメージは結局そういったものから独立していることになるのです。

面白さは見えるか？

つまり，面白いゲームというのは，直接ゲームから面白さが与えられるのではなく，ゲームが返してきた反応をプレイヤーが快く理解できるゲームということになります。だとすれば，それを生み出している，プレイに対するゲームの反応こそが，もっとも重要な因子であることは明らかです。ではゲームから与えられる反応とはいったいなんなのでしょうか？

それを考えるために，とりあえず自分がゲームをしている状況を想定してみましょう。マウスやジョイスティックを操作してゲームに自分の意志を伝えれば，それが画面や音の変化によってプレイヤーである自分に返ってきます。これは具体的な動きの違いこそあれ，どんなゲームにも共通なものです。ゲームセンターの大型ゲームでは椅子の動きや画面以外のイルミネーションなどで，さらに派手な反応を返してくれることもありますが，結局はどれも表面的な違いでしかありません。

するとその反応のなかに含まれた情報に，その本質があると考えられます。操作に対する変化に対して我々がとる態度は，それに対して満足するか，否かということに集約されます。つまり欲求と充足が存在しているわけです。そう考えると，光や音を使ってゲームが我々に伝えているものは，欲求を満たしたときの「快感」ということになるのではないでしょうか。

非常に大雑把なことをいってしまうと，人間が不快なことを自発的にやったりすることはありえません。行動の動機にはなにかしらの欲望があり，欲望が満たされることで快感が生まれます。つまりゲームも，こういった行動の原理に従っていると考えられるわけです。

これはゲームの外観や形式からだけでは得られない性格のものです。ゲームの面白さを左右できるものが，見えない存在であるということは，とりも直さず面白いゲームを作ることの難しさを示していることに，ほかならないといえるでしょう。

行為と代償

だったら気持ちいいゲームを作れば面白いゲームになると，考えるのは間違いではありません。では，気持ちいいゲームとはどんなゲームかと考えても，具体的になにひとつ見えてこないのでは，ここまでとなんら変わっていないことになります。そこで先ほど説明したことを，もう一度考え直してみましょう。

快感が得られるためには，目的をともなった行動が必要です。欲求や欲望を持ち，目標を持って行動することに対して，その行動の結果の良否が快感を左右しているわ

けです。ということは，このあらゆる行動の目的に見合う結果を，漏らさず十分に用意してやることが，快感を生むといえるのではないでしょうか。自分のやりたいことができ，それに満足のいく反応が用意されている，これこそが，面白いゲームの条件ではないのでしょうか。

たとえばシューティングゲームなら，弾を撃つことが，最大の目的行為であり，ボタンを叩いただけ十分に弾が出ることが，その結果であるはずです。アクションや格闘ゲームなら，操作に素早く反応し，自在にキャラクターを操れることになります。これらは，比較的単純な行為によって小さい快感が与えられるタイプになります。よって何度も快感を得られるために，プレイの時間を維持することで快感を持続できるわけです。

ほかには，ロールプレイングやシミュレーションゲームのように，戦闘や移動の行動を起こすことで，シナリオで掲げられた目標への接近が結果として与えられるものもあります。この場合は，そうした結果そのものは快感ではなく経過であることがほとんどで，目標やイベントが終了したときに，大きな快感が得られるものが多いのが特徴です。これは行為と快感の周期が大きい分だけ，その規模が大きくなっているといえます。

こういった規模の差こそあれ，目標→行動→結果→快感というシステムが成り立つことで，ゲームは成り立っているのです。面白いかどうかは，このゲームシステムがいかに緻密に組み立てられているかに，大きく依存していることは，いうまでもないことでしょう。

快感の違い

先に例を挙げただけでなく，ゲームから得られる快感は，実際にはもっと多岐に及んでいます。流行のバーチャルリアリティを意識したゲームでは，自分の操作する環境や場所自体も特殊なものが用意されており，そこに入ることですでに十分な快感を得られる場合があります。綺麗なデモ画像を見ているだけでも，うっとりするようなゲームも確かに存在します。音楽が好きで好きでたまらないゲームなどがあっても，別段珍しくもないでしょう。

これらの例は快感が形から提供されているので，いままでの話と食い違っているように思えるかもしれません。しかし，これらはゲームシステムの快感とは違い，付加された別な種類の快感だといえます。すなわち，いくら綺麗なグラフィックや雰囲気バッチリのコックピットが，ゲームを飾り立てていても，それらはゲームに対する行動の代償としての面白さではないわけです。

確かに十分魅力的で，プレイヤーに快感を与えてはくれますが，最悪の場合，ゲームシステムから快感が得られないときのごまかし役を担うだけになってしまう危険性があります。そういった形からだけの快感に注目するあまり，ゲームシステムがおろそかにされた「曲だけはよい」とか，「グラフィックデモのために……」などという中途半端なゲームはあとを絶ちません。

快感が得られるからといって，必ずしもそれが面白さに直結してはいないことには，注意が必要でしょう。面白さは快感ですが，逆は必ずしも成り立たないのです。

しかも，直接の行動を必要としない快感は，いつでも得られるその性質のために，早々に快感の価値が失われる可能性を否定できません。つまり「飽きてしまう」わけです。そうなった場合，ゲームシステム自体から快感が生み出せないとすると，そのゲームは，その存在すら危なくなってしまうことになります。快感の価値は，そのままゲームの価値でもあるのです。

面白さの価値

ゲームはプレイヤーの意志さえあれば，何度でも反復することが可能です。これは何度でも快感を得られるということである反面，そうすることによって同一の快感にプレイヤーが慣れてしまうという問題があるわけです。結局，快感の価値の低下はここでも起こるのです。

しかしこの問題は，ゲームのシステムを複雑にすることで解決できる性格のものです。つまり，目的行為に条件を追加し，それを満たした場合に，より大きな快感を与えるということです。

たとえば，敵の攻撃を避けるといったような単純なものから，特殊な条件を満たして目的を達成した場合に対するボーナスポイントなどが例に挙げられます。こういったことによってシステムをより強固に組み上げた結果，俗にいわれる奥の深さとか，飽きのこない面白さというものが獲得されているのです。

ここで注意しなくてはいけないのは，あくまでも単純な目的行為に対して，付加的な「おまけ」として，条件を加えていかなくてはいけないということです。目的に反したり，それとは掛け離れた行為を脈絡なく快感に結びつけても，それが問題の解決には決してなりはしないからです。別な遊び方ができて，本来のものより優れた快感のシステムが成り立つとしたら，それは本末転倒ということになってしまいます。

こうしたなかでシステムを組み上げ，形からの快感を融合させることで，新たな面白さが生まれてくるかもしれません。そうしたあらゆる選択は，すべて開かれており，ひとえに作る側の自由に任されているわけです。だからといって，これは個人のセンスとかフィーリングというような，形のない感覚に依存しているわけではありません。実際にゲームの形になった場合に，それが本当に快感になるのかを，きちんと検証さえすれば，必ず面白いゲームになるはずだからです。最初に挙げた，面白いゲームの模倣という方法においても，その快感さえ理解していれば，負けないくらい面白いゲームを作ることができるはずなのです。

いつでも楽しく遊びたい

このように，快感を得られるシステムを組み上げることがゲームを作るということであり，これはデザインの段階だけでなくあらゆる場面に意識されるべき課題です。

ここでは，形から与えられる快感も含め，それらを管理することで，ゲームの面白さが自在に操れるものであることを示してきました。しかし，現実には自分の能力や，時間といった要素などを無視することは困難で，これは机上の空論なのかもしれません。しかし，面白いゲームを作りたいと思うなら，努力や作業のなかで，このことを忘れてほしくないと思います。ゲームは，みんながいつでも楽しめるものなのですから。

リアルタイムゲームのための

“基本”スプライトルーチンの作り方

Yokouchi Takeshi 横内 威至

X68000のゲーム作成の際にもっとも基本となるのはやはりスプライトです。ここではスプライト表示システムの基本的な構造から，ゲーム作成時のテクニックまで一挙に紹介します。

ゲーム特集ということで当然スプライトを扱うのである。X68000のスプライトは最高128個しか表示できない。まして拡大縮小だとか回転なんてものはついていない。アーケードゲームと比べると時代に取り残されている感はあるが，やはりシューティングゲームなんかでは重宝する。8バイト分書き込むだけで16×16ドット，さらに重ね合わせもやってくれるからだ。グラフィックを使うのとでは明らかに差が出る。X68000自体そんなに速いマシンではないので，このことはスプライトを使うことの最大の利点となっているのはみんな知っているはずだ。

このように便利なスプライトを正しく管理することは多くのゲームにとって非常に重要なことである。ということで，今回はスプライトの基本的な操作方法を探ってみようと思う。

ハードウェア

まず，どんなことができるか理解しておこう。8月号の村田氏の連載でしっかりと説明されているので，細かいことはいまさらいわないことにする。BGについては今回は使わないので無視。ピュアにスプライトだけを押さえておいてほしい。

スプライトにはまずPCGエリアというのがある。ここは16×16ドットのパターンを256個定義しておける。ただしBGを使わなければ，だ。BGは実はそんなに役に立つものではないので，あまり使われていないのである（半分ウソだが）。あとで特殊な使い方を示そうと思うのでいまはパス。

で，BGを使うときはスプライト専用となるエリアは最低で64個分まで落ちてしまう。BG1枚だったらスプライト専用に128個使えるが，スプライトの最大表示数が128であることを考えるとやはり厳しい。

動いている物体はいろいろとアニメーションしているのが常なので，ヘタしたら毎回このPCGを書き換える羽目になる。さすがにいちいち書き換えていてはスピードは苦しくなってくる。

例によってG2なんかはがんばっているが，スピードとしてはギリギリである。15kHzモードだと一瞬キャラクタがブレることがある。つまりこれはどういうことかというと，このPCGを書き換えているあいだにCRTCは表示期間に入ってしまっているということだ（15kHzのほうが31kHzより垂直表示期間がやや短い）。

長くなってしまったが，次へ。次はスプライトスクロールレジスタだ。これは128組，つまりスプライト128個分ある。16×16ドットの画面が128枚あると思えばよい。当然優先順位は0＞1＞2＞……＞127となっている。細かいことは村田氏の連載（7，8月号）のほうを見てほしい。

ゲームへのアプローチ

実際にいろんなゲームではどうやってこのスプライトを制御しているか考えよう。ゲームに必要な要素はたくさんある。入力だとか計算だとか判定だとか，そして表示だとかだ。まあとりあえず簡単なシューティングを考えよう。

まず自分の制御だ。ジョイスティックなどで入力し，座標を計算してバッファに置いておく。弾なども計算，判定してバッファに置く。敵もまた計算，判定してバッファに置く。もちろん弾が当たればいろいろ処理したりもする。そして背景なんかもスクロールする座標を計算して，やはりバッファなんかに置く。と，同時に次の背景を書き込んだりする。さて，そして表示だ。ここが今回のポイントとなる。

ゲームの場合，処理によってスピードが変わることはあまり許されない。ということは一定の周期でひととおりの処理をすることになる。これはご存じのとおり，画面の書き換えを行うときにチェックするのが定めだ。

CRTの動作は，画面左上から書き始めて右端まで通して走査する。そしてちょっと時間をかけ，下の段の左端からまた走査を始める。そのまま同じく最下段を書き終えると，しばらく（といっても1.7ms程度）時間をかけてまた最初に戻るのである。これは常に一定周期で行われており，これに合わせてスクロールなんかのレジスタを書き換えてやればよいのである。

この説明で最後の期間（垂直帰線期間という）に書き込めば，画面の上から下まで変わらぬデータで表示をすることになる。ちなみに，途中の“ちょっと時間をかけ”のときにスクロールを書き換えてやれば次の段からスクロール表示が変わり，ラスタースクロールというやつになる。

で，垂直帰線期間に書き込むのだが，まあ，そんなに時間はない。せいぜいスクロールレジスタ分のブロック転送をしてやるのがいいところだ（本当はもうちょいと余裕はあるが）。だから，それまでに書き込むデータの写しを別のアドレスに作っておいてやるというのが筋とされている。

話は戻るが，ゲームの周期調整を行うのがここである。いったようにいろいろな処理を終え，スクロールレジスタに書き込むデータを作っておき，この垂直帰線期間に入るのを待ってやればOKだ。そうして表示してやり，また最初から継続していってやるのが一般的な方法である。

簡単にまとめると，垂直表示期間には一切表示に手をつけず，移動なんかの計算に専念する。そして垂直帰線期間にドバッと表示部分を書き換えてやるということだ。最初にバッファがどうだとうるさくいったが，つまりこのバッファを元にスクロールレジスタのデータを作るのがカギとなっていることはわかってもらえたと思う。

ゲームっぽいサンプル

ちょっと長めになっているが以上のことを踏まえたサンプルを用意してみた。応用だとか問題点はあとに回すとして，とりあえずこのリストの解説をしてみる。

256×256，高解像度の画面でグラディウスのオプションの腐ったような敵を動かすサンプルである。敵のくせにジョイスティックかカーソルキーで動いてしまうイカれた奴らだ。画面中央やや右下あたりはブービートラップがあり（画面に表示はない），そこに当たるとよくあるゲームのように色反転するようになっている。スペースキーを押すと終わりにしてある。

●データフォーマット

まず本題ではない敵まわりを解説する。敵1匹あたり64バイトのデータを与えてやっている。1ワード目はフラグ。0で空きエリアを意味する。ビット14はダメージを食らったときに1になるフラグ。ほかのビットは今回のサンプルでは無視。

2，3ワード目はまとめてロングワードとなり，X座標を示す。特に，下位ワードは微調整用の小数点以下を示すこととなっている。

4，5ワード目は同じくY座標。この座標はいろんな判定などを助けるためにだいたい1024+αの値となっている。これはどういうことかというと，0+αだと画面左や上に敵が消えたときには座標がマイナスになってしまい，そのほかのものとの位置関係の計算がややこしくなるのでこうしている。どうせスプライトスクロールレジスタは1024を示すビットは無効だからこれでよいのである。あくまでもひとつの方法としてとらえてほしい。

6ワード目は判定用のX方向の長さ。7ワード目は同じくY方向の長さ。サンプルではまったく使用していない。8，9ワード目はその敵の移動用プログラムのアドレス。10,11ワード目はそいつが死んだときのプログラムのアドレス。このサンプルでは死ぬことはないからあまり意味はない。

12，13ワード目はグラフィックパターンのポインタがあるアドレス。あとで説明する。14ワード目はアニメーションパターン。15ワード目は耐久力。16ワード目はダメージを受けたときのそのダメージ。

17ワード目はプライオリティ。BGとの前後を示すヤツだ。そのままスクロールレジスタに書き込む値である。それぞれの敵との前後は関係ない。ちなみに敵同士のプライオリティはバッファアドレスの低いほうが優先されている。18ワード目からはそれぞれが勝手に使えるワークエリアとなっている。このオプションたちはXYをテーブルから得るための相対ポインタとして18ワード目を使っている。

次に，結構重要な部分，敵のグラフィックパターンについて説明しておく。先ほどの12，13ワード目のアドレスには，さらにポインタがいくつか書かれている（リスト352行）。このポインタ（ワードサイズ）を，14ワード目の数値をパターン番号としてそのアドレスに足してやるとその敵のグラフィックパターンを示すデータのアドレスとなるのだ。わかりづらいからこれはリストの“SETENEMYSPB”をよく見てほしい。

で，このグラフィックパターンを解説する。最初のワードはそいつがそのパターンで使うスプライトの数である。今回はみんな4個ずつ使っている。そしてその先は3ワードで1組であり，その最初はX座標のオフセット値，次がY，最後はそのままスプライトスクロールレジスタの3ワード目に送るデータ，反転，パレットテーブル，スプライトコードのデータである。これが今回は4個ずつそれぞれのパターンにある。このオフセット座標を使えば矩形のキャラクタ以外にもいろいろな奴，たとえば十字型なんかもできる。

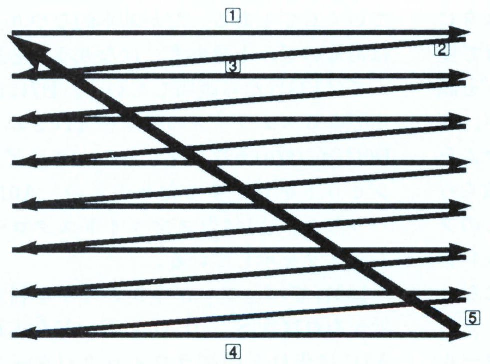
そして，判定領域をキャラクタの中央だけにしたりとか一部にしぼったりもできるので，普通はこうしているのである。まあ，いろんなフォーマットがあるからあくまでも一例にすぎないことは覚えておこう。

プログラムの説明

さて，では流して解説しよう。

まず最初は見てのとおり。初期化だ。34行目からは敵データを作ってやっている。XYテーブル上のどこを使うかを敵ごとにずらしてやっているだけ。あとはアニメーションパターンをちょっとずつずらしてやっているだけ。次はスーパーバイザモードに固定してやる。本当はあまり感心できない

図1　CRTの動作のいい加減な説明



①左から右へ走査
②次の段（ラスター）の左端へ
③再び右へ走査
⋮
④最下段走査
⑤左上へ

①～④：垂直表示期間
⑤：垂直帰線期間

ゲームなんかでは①～④の期間（実際はちょっと⑤の後半も入るが）に計算等をすませ，⑤の期間に入ると同時にスクロールレジスタなんかを書き込むことにより周期を保ち，そして画面の乱れを防いでいる。

ちなみに②のような期間にスクロールレジスタを書き換え，その上下のラスターでスクロールに差をつけるのがラスタースクロールというやつである。

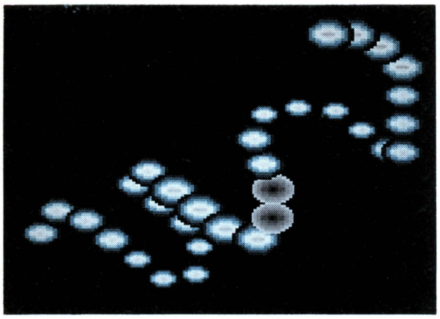
図2　敵データのフォーマット

オフセット（ワード）
- 0：フラグ。①だと空き。特に第14ビットはダメージフラグ
- 2：X座標（ロングワードで2ワード目は小数点のような役割）
- 3：Y座標
- 5：X方向の判定ドット数。未使用
- 6：Y方向の判定ドット数。未使用
- 7：移動プログラムアドレス
- 9：死亡プログラムアドレス
- 11：グラフィックパターンアドレス
- 13：アニメーションパターン
- 14：体力
- 15：ダメージの大きさ
- 16：プライオリティ。0は表示しない
- 17：ワークエリア

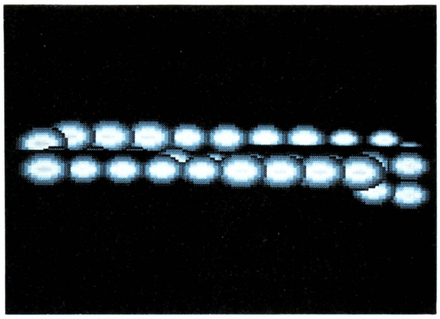
図3　グラフィックパターンのフォーマット

[N]，［X_1，Y_1，SP_1］，［X_2，Y_2，SP_2］……［X_N，Y_N，SP_N］
N個

最初のワード，Nは使用するスプライトの数。次いでX，Y，SPの組がN個。バッファでのXYにこのX，Yを足した座標にSPで示されたスプライトを表示する。



サンプルプログラム



スプライトが消える

が，いちいち変えるのは面倒だし，市販品クラスの大きなものは皆このままで動いているに決まっているのだ。

51行からはパレットの設定。テーブル14，15を使う。テーブル15はダメージ時のフラッシュ用だ。この程度のことでわざわざIOCSコールは使いたくない。そのあとはスプライトPCGを定義してやる。$40から$42までを使っている。大，中，小のそれぞれ左上部分だけを作ってやっている。あとのところは先ほどのグラフィックパターンのところを見れば反転によって表示されることがわかるだろう。面倒なら適当なグラフィックを描いて使ってももちろんかまわない。

その後ろはいよいよメインルーチンだ。といってもメインルーチンを呼び出す部分をコール，戻ってきたらスペースキーをチェック，それによってループするだけだが。終わったら設定を戻して終わり。ということでさらに深みへ入る。

1：ENEMYMOVE

敵を動かすところ。まずA6レジスタに敵バッファ先頭を入れておく。209行で空きかどうか調べる。211行でダメージを調べる。そのあと死んだかどうかを調べ，死んだら移動プログラムと死亡プログラムを替えてやる。で，生きててダメージを食らったならフラグを立てておいてやる。このフラグはあとの仮スプライトスクロールレジスタのデータを作るときに使うことになっている。そして，A6にその敵のエリアの先頭を残したまま移動プログラムをコールしてやる。あとは敵の数だけループということになっている。

なぜデータを読んで操作するインタプリタにしないかというと，もちろんこのほうが応用がきくからだ。これならそのデータインタプリタをコールすることもできる。敵の動きを多様化させるにはこれがもっともよいと私は勝手に思っている。もっとよい方法があるとも思うが。

ついでに敵プログラムも説明してしまう。簡単だからあまりしないが。敵1は頭のやつ。プログラムは274行からとなっている。入力をしてもらってXYを計算するだけ。動いたらオプション用XYテーブル(269)行を動かしてやる。291行をカットしてやれば動かなくてもオプションが動くようになりツインビーみたいになる。敵2はポインタの指示どおりにテーブルから座標を拾ってくる。その座標が一定範囲内ならダメージは0だけど判定フラグを立ててやる。ちょっと邪道なやり方だがダメージを被ったのと同じこととなる。あとはアニメーションをカウントしてやっている。1周期ごとにアニメーションではせわしないので別のカウンタを使って操作。以上。

2：SETENEMYSPB

ここがポイント。敵バッファから仮スプライトスクロールレジスタにデータを作ってやるところである。例によって，あくまで方法のひとつである。今回のフォーマットがこうだからこうしているだけであり，フォーマットが違えばもっとほかの方法だっていくらでもある。だが基本的なやり方ではあるので，十分参考にはなりうると思う。

では133行から説明していく。D7には使用できるスプライトの残りを入れておく。D6は敵の数のカウンタである。D5はダメージビットを判定するためのもの。A0は敵バッファ，A1は仮スプライトスクロールレジスタを示している。

139行でバッファが空かどうかを判定。敵があれば次へ。D1,D2にX，Yを入れる。A2には先ほどのグラフィックパターンポインタ，A3はデータレジスタが足りないため，代わりにプライオリティを入れておく。そしてD0にパターンを入れ，A2にグラフィックデータアドレスを計算してやる。

そしてグラフィックデータをやっと扱ってやることになる。使用スプライト数を拾い，D7を計算してやる。このとき，オーバーしたらその敵は表示しないことになっている。その際，D7は戻してやり，再び次の敵を扱ってやっている（153行）。さて，154行だがD0をループカウンタとするため，−1してやる。DBRA(DBF)は0で1回ループを通るためだ。当たり前だ。

そのあと，ダメージを食らったのとそうでないのとに分かれる。両者の違いはほぼなく，ただスクロールレジスタのパレットブロックの差だからダメージを受けているときのほうを説明する。159行から表示すべきスプライトのオフセット座標を計算して仮スプライトスクロールレジスタに書き込む。続いてSPコード，パレットテーブルを含むワードを拾い，166行でパレット15にしている。それを書き込み，そのあとA3，つまりプライオリティを書き込んでいる。あとは使用する分だけループして繰り返すだけだ。

そのままスプライトの数を超えなければすべての敵についてこれらを行ってやることになる。最後に，189行からは残った分のスプライトについていじってやる。スプライトスクロールレジスタの8n+6バイト目がプライオリティ，これが0なら表示しないので，余ったスプライトはこれをすべて0にしてやる。

以上で仮スプライトスクロールレジスタの操作は終わりである。

3：WAITDISP

106行からのところ。当然これは垂直帰線期間になるまで待つものだ。割り込みで行ってもよいが，そうする必要もない。MFPのGPIPデータレジスタ（$E88001）のビット4を調べればいい。これが0だと垂直帰線期間，1なら垂直表示期間であることを意味している。110行で，一応表示期間でこのルーチンに入っていることを確認している。

もしこの確認をしないと，メインの処理が速すぎて帰線期間内に終わったとき，表示前にもう一度処理してしまうことになる。またメインが遅すぎて，すでに帰線期間後半ぐらいに，あるいは表示期間直前にこのルーチンを通ると垂直表示期間にスクロールなどを書き換えることになり，画面ブレなんかを起こしてしまうことになる。それだけならいいがラスタースクロールだとかを行うとするとさらに不都合が起こることになる。

そのためにまずこのルーチンに入ったときが垂直表示期間であることを確認している。そして114行で垂直帰線期間になるまで待つのである。たったこれだけだ。簡単なものであろう。だから，処理がゲーム中に重くなり，1周期で処理できないときは2周期分の時間がかかってしまい，スピードが極端に落ちてしまうことになる。いわ

ゆる処理落ち，スローなどというやつだ。いろいろなゲームで体験しているに違いない。

4：MONITORON

一気にスプライトスクロールレジスタに書き込むルーチンだ。

まず120行でスプライトの表示をオフにしている。スプライト関係のレジスタは読み書きできるポートがひとつであり，CPUとCRTが同時にアクセスすることができない。CRTに優先権があるため，CPUはウェイトが入ることになってしまう。どうせ帰線期間は表示なんか関係ないからCPUにウェイトが入らないよう表示をオフにするのは鉄則とされている。あとは122行からは本物のスクロールレジスタに転送してやるだけである。最後にはまた表示をオンに戻してリターン。本来ならここのループは展開しておく。つまらぬことで時間を食うのはリアルタイムゲームにとっては致命的なことだからだ。

さらに応用を考える

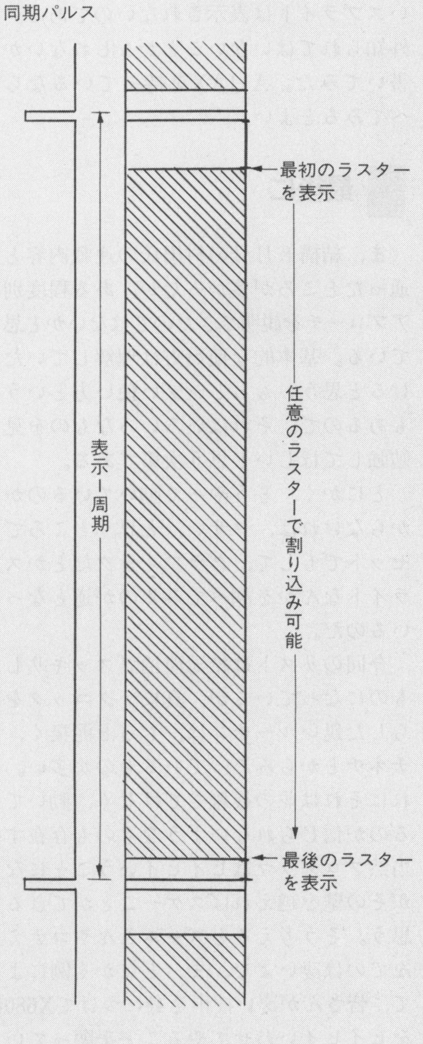
ひととおり基本は押さえておいたつもりである。だが動かしてみるとまずひとつ気がつくことがあるかもしれない。キャラクタがうまく表示されていないときがあるのを見ていただけただろうか。横に直線に並べるとケツに近いほうの奴が表示されないことに気がついただろうか。これはハードの制約のひとつである。スプライトは横32個以上は表示してくれないのだ。

で，もし32個以上並べるとどのスプライトが表示されないのだろう。見ればわかるし，考えれば見当がつくだろう。そうである。スプライト番号の高い，つまり優先順位の低い奴は無視されることになっている。では絶対この消えた奴を表示することはできないだろうか。

とりあえず方法はある。単純に優先順位を毎回変えてやればよい。表示ごとに消えてしまうスプライトをずらしてやれば，ちらつきは出るが，まあ一応表示はしてくれる。一応これも変更用のサンプルを用意した。“SETENEMYSPB”をリスト2のようにすればいい。あまりよい方法ではないので，サンプルでは重ねると異常なまでにちらついてしまう。ただ，改良点はいくらでもあろう。たとえば優先をキャラクタごとに決め，常に優先されるもの，そして消えてもよいものに振り分け，それによってスクロールレジスタを作ってやってもよい。

これは今回のサンプルの大きな弱点，キャラクタの優先順位そのものにもかかわる。このフォーマットでは敵エリアのアドレスの低いほうが常に優先される。これはゲームによっては致命的なことになる。

180300クロックの死闘



CRT（要するにディスプレイ）は，電子銃から放たれたビームを上から順に走査することで平面の絵を表示する装置である。実際には1ラインずつしか表示していないのだが，それでもちらつかないのは蛍光体の残光による効果が大きい。画面表示というものが，大きな意味を持つゲームの世界では，このCRTの基本構造がプログラムに大きく影響するのだ。

左の図は，CRTC（CRTコントローラ）関連の表示アクセスタイミングを表しているものだ。通常，横に描かれるものを縦方向にして画面走査に対応させやすくしてみたものである。各部の長さの比率もだいたい実際の時間間隔に一致するようになっている。

図中の斜線がかかった部分が画面表示期間，そうでない部分は画面になにもアクセスしていない期間を表している。画面制御でよく聞かれる「垂直帰線期間」という言葉はおおむねこの非表示期間のことを指している。

しかし，見ればわかるように，画面表示期間のほうが圧倒的に長い。また，画面の上のほうを書いているときと画面の下のほうを書いているときでは，これだけの時間差があるということも覚えておいてほしい。

たとえば，表示期間中に瞬間的に画面を書き換えたとしよう。CRTが1画面分の内容を書き終えた時点では，ある走査線を境にして上下の内容がまったく違うものになる。これは次の周期には解消されるものである。しかし，画面の途中であることと「次の周期で解消される」ことが画面ちらつきの原因となる。

これを防ぐには，画面書き換え（スクロールやパレットの変更，スプライトの書き換えを含む）をすべて画面表示していないときに行うしかない。

垂直帰線期間は画面のいちばん下まで走査し終わった電子銃を画面の上にセットし直すまでの期間である。画面表示のための基本設定処理はこの期間に行われる。

通常，リアルタイムゲームは1/60単位で作成されている。というか，1秒間に約60回行われる画面の書き換えにタイミングをあわせて動作するように作られる。ちなみに，ディスプレイを31kHzで表示しているときの表示1周期は18.03m秒となっている。15kHzのときは16.27m秒である。

通常のX68000ではCPUが10MHzで動作している。ということは，とりもなおさず1秒間に10,000,000クロック分の命令を実行できるということだ。しめて180300クロック以内の命令でメインループを構成すればよい（実際には割り込みなどの影響も考慮しなければならないので多少異なる）。

一般的な構成では，画面周りにアクセスできない表示期間中にさまざまな計算などの処理を済ませておき，垂直帰線割り込み（MFPで同期信号の立ち上がりを指定）で必要なデータを一気に書き込む。スプライトの書き換えなどもこのときに行う。垂直帰線期間は31kHz時約1.8m秒である。この時間をいかに使うかが重要である。

グラフィックやテキストなどで2ページ以上の画面が持てる場合は，2ページ構成にして，表示していないほうに書き込み終わってからページ切り換えやスクロールによって（これは垂直帰線期間に行うこと），表示期間中でもなにも気にせずにデータの書き換えを行うことが可能である。

ちなみに，同じラインの走査線が左端から右端まで行き着くのにかかる時間は22.09μ秒。220CPUクロック分の時間となる。この期間に処理を行うのがラスタースクロールなどである。このあたりの処理になると，CRTCなどのレジスタを書き換えたあと，その影響が画面に現れるまでどのくらいかかるかといったことまで考慮しなくてはならなくなる。

また，指定したラスター表示で割り込みをかけられるので，表示ラスター数を減らして（スプライトを表示OFFにして）時間を稼ぐなどという技も使えるかもしれない。

たとえば，ファイナルファイトなんかではどうだろうか。手前のものが奥の奴に隠れてしまったりすることになる。毎回これらのバッファを入れ替えるのもよいし，またはデータとして優先順位を持たせてもいいし，またこのバッファをただ順に読むのではなく，DMAのリンクアレイチェイニングのようにポインタを使って読んでやるのもひとつの手である。

ハードウェアの制約として1ライン表示個数の限界（32個）以外に画面中での表示個数の限界（128個）という問題がある。アーケードゲームでのン千個に比べると頼りないが，家庭用ゲーム機などと比べると，実はそんなに少ないほうではない（というよりかなり多い）。

しかし，それらのゲームを見ると実によく動いているのがわかると思う。

理想的には無限個のスプライトがあればいいのだが，スプライト個数を増やすと，スクロールレジスタやなんやら，管理に必要なメモリなども膨れ上がってしまう。それで限界は必ずできるものだが，これは実は一度に処理できる個数であって，1画面での制限ではない。わかりにくいと思うが，1画面描くのに1回スプライトを表示させているのを，1画面で2回にすれば表示個数は倍になる。

具体的には，上のほうで使ったスプライトを下のほうで，もう一度使うという技が可能なのだ。CRTの表示はラスター（走査線）ごとに行われている。スプライトはこの表示と同期して働く表示システムだ。ラスターごとだから，関係あるのは現在処理中のラスター近辺だけ。極端にいえば，処理ずみの部分は20ラスター上でも3秒前の画面でも同じことなのだ。これもうまくシステムにすれば表示個数は格段に増加させることができるだろう。

このようにハードウェア上の制約を突破する手もなくはないのだが，市販ゲームでは素直に制約を受け入れ，パターン書き換えとテキストやグラフィックなどを併用することでうまくかわしている。「限界」内でもかなり高水準のものが作れることがわかるだろう。

改造するべきところはまだまだあるはずだ。グラフィックパターンも固定ではなく，もっと自由度を高めるためにポインタを使ってやれば触手なんかも簡単に作れるだろう。パレットも固定せず，キャラクタごとにデータを持たせてやるのもよい。まあとにかく内容によってより適したフォーマットを作るのが肝心なことであるということを覚えておくことだ。しつこいがあくまでもこれはサンプルで，ほんの一例にすぎないのだ。

もうひとつ，MONITORONルーチンのところで考えてみることがある。転送が終わったところでそのまま表示オンにしているがこれは明らかにムダなことである。

またもやG2のことであるが，これらはおそらくスクロールレジスタの書き換えが終わってすぐにスプライトPCGを書き換えているか，あるいはスクロールレジスタを書いていると同時にDMAでPCGを書き換えているはずだ。ここのアクセスもやはり素早く行いたいものである。

ただし，それなりに時間がかかるものであり，帰線期間内に終わらないかもしれない。まあDMAを使って書き換えてもいいし一応なんとかなるだろうが，やはり表示ギリギリまでは表示オフにしておいたほうが得であろう。だから，ここでラスター割り込みを使ってやるとよりイカした高度なプログラムになる。ラスター0の割り込みを使い，その瞬間に表示をオンにしてやると結構テクニックぽくてかっこいいような気がしないだろうか。まあ，これは自由研究としておこう。これくらいは自分でやってみたほうが勉強になると思う，といって逃げておこう。

さて，余談ではあるが最初のほうでG2が15kHzでブレることがある，といったがつまりこれは表示開始後までこのPCG書き換えがかかっているということである。ここでちょっとつまらぬことを考えてみた。G2はBGも使っているからダメだけど，BGを使わないときは128個分のPCGが余っている。よって0から127，128から255までを分けて1周期ごとに交互に書き換え，書き換え中でないほうを表示してやればこのようなキャラクタグラフィックの狂いは防げる。これももしかしたら使えるテクニックではないだろうか。あ，いっていることがよくわからないかもしれない。格闘モノなんかでPCGが足りないと思ったとき，参考になるかもしれないからそのときは考えてみてほしい。まあいいや。

本題からはずれてはいるが，せっかくだからBGの隠れた(?)使い方があることをいっておこう。グラフィック，テキスト，スプライト＋BGはそれぞれのあいだでプライオリティが決定できることは知っていると思う。そこで，よくある縦スクロールシューティングを考えてほしい。まずテキストは普通点数なんかの表示のために使うからもっとも優先すると思う。次はスプライトだ。そしてグラフィックであろう。

ただ，究極タイガーなんかを考えるとどうだろう。建物なんかに敵が隠れていることがあるだろう。この建物はBGで描けば問題はないが，16色であること，パターン数が少ないことを考慮すると大きいものや複雑なものには不利となる。ということで，

　BG＞スプライト

かつ

　グラフィック＞BG

それでいて，

　スプライト＞グラフィック

となることがあれば，建物のグラフィックを覆うことができるパターンをBGに用意すればよいことになる。

優先はBG＋スプライトがひとまとめであることを考えると不可能であるように思えるが，実は可能なのだ。ただ単に，BGの使っている色のRGBIを0にしてやるとこうなるのだ。ただしカラー0は本当に透明だからダメ。使っているパレットテーブルのカラー1から15を0にしてやると，下のグラフィックは見えるがBGより優先が低いスプライトは表示されないのである。案外知られてはいないことかもしれないから書いてみた。A-JAXを持っているなら調べてみるとよい。

最後に

ま，結構8月号の村田氏の連載内容と似通ったところが多かったが，ある程度別のアプローチを説明できたのではないかと思っている。基本的な使い方は理解していただけると思う。もっとシブい使い方というのもあるので，それはいろいろなものを見て勉強してほしいと思う次第である。

とにかく，どうやって動いているのかわからなければ，クサい，と思うところでリセットでもして，グラフィックだとかスプライトなんかを調べてみるのが道となっているのだ。

今回のリストは結構淡泊でスッキリしたものになっているが，真にテクニックを凝らした鋭いルーチンは，もっと泥臭く，ネチネチとからみあっているものが多い。それはそれは並の複雑さではなく，動いているのが信じられないようなものも存在する。当然プログラマはヒイヒイいうことになるがその壁を越えればスゲーことができると思う。そう考えるとズームさんやコナミさんてのは凄いよな。ま，とにかく例によって，皆さんが凄い技術を身につけてX68000をヒイヒイいわせてやることを願っていよう。

リスト1

```
	.include	DOSCALL.MAC
	.include	IOCSCALL.MAC

	.offset 0
FLAG:		ds.w	1
ENEX:		ds.l	1
ENEY:		ds.l	1
ENEXL:		ds.w	1
ENEYL:		ds.w	1
MADR:		ds.l	1
DADR:		ds.l	1
GRADR:		ds.l	1
PAT:		ds.w	1
VITAL:		ds.w	1
DAM:		ds.w	1
PRW:		ds.w	1
PAREA:		ds.w	1

	.even
	.text

	move.w	#-1,d1
	IOCS	_CRTMOD
	move.l	d0,crtfst
	move.w	#$100+10,d1		*256*256, 256色*2, HI
	IOCS	_CRTMOD

	moveq.l	#2,d1
	IOCS	_B_CLR_ST		*テキストクリア
	IOCS	_B_CUROFF		*カーソル非表示
	IOCS	_SP_ON			*スプライト表示
*
	moveq.l	#29,d0			*敵データ作る
	lea.l	enemy2(pc),a0
	lea.l	enemy3(pc),a1
sete1:
	moveq.l	#16-1,d1
sete2:
	move.l	(a0)+,(a1)+		*1キャラ64バイト
	dbra	d1,sete2
	add.w	#16,PAREA(a0)		*データ上での本体との距離
	move.w	PAREA+2(a0),d2		*パターンをずらす
	addq.w	#1,d2
	and.w	#15,d2
	move.w	d2,PAREA+2(a0)
	dbra	d0,sete1
*
	bsr	SUPER			*スーパーバイザー固定
	and.b	#$0f,$e9a005		*ジョイスティック初期化
	lea.l	PALET(pc),a0		*パレット書き込み
	lea.l	$e82200+32*14,a1
	moveq.l	#31,d0
sete3:
	move.w	(a0)+,(a1)+
	dbra	d0,sete3
	bsr	SETSPRITE
*
loop:						*メインルーチンです
	bsr	MAINLOOP
	moveq.l	#$06,d1			*スペース?
	IOCS	_BITSNS
	btst.l	#$05,d0
	beq	loop			*でなければループ
*
	bsr	USER			*ユーザーモード固定
	move.l	crtfst,d1		*画面モード戻す
	IOCS	_CRTMOD
	IOCS	_B_CURON		*カーソル表示
	DOS	_EXIT
*
*
*==============================
MAINLOOP:
	bsr	ENEMYMOVE		*敵動かす
	bsr	SETENEMYSPB		*敵データから仮スプライト
					*データを作る
	bsr	WAITDISP		*垂直帰線期間を待つ
	bsr	MONITORON		*仮スプライトデータを書く
	rts
*
*
*==============================
SUPER:
	move.l	d0,saved0
	move.l	a1,savea1
	movea.l	#0,a1
	IOCS	_B_SUPER
	move.l	d0,sspbuf
	move.l	saved0,d0
	move.l	savea1,a1
	rts
USER:
	move.l	d0,saved0
	move.l	a1,savea1
	move.l	sspbuf,a1
	IOCS	_B_SUPER
	move.l	saved0,d0
	move.l	savea1,a1
	rts
saved0:		dc.l	0
savea1:		dc.l	0
sspbuf:		dc.l	0
*
*==============================
WAITDISP:
	lea.l	$e88001,a0		*MFP
	moveq.l	#4,d0			*GPIP4 V-DISP
waitdisp1:
	btst.b	d0,(a0)			*まさかまだ帰線期間だと
	beq	waitdisp1		*早すぎるから一応チェック
waitdisp2:
	btst.b	d0,(a0)
	bne	waitdisp2		*表示期間終わりまでループ
	rts
*
*==============================
MONITORON:
	lea.l	$eb0808,a2		*帰線期間中は非表示にする
	moveq.l	#$20,d0
	eor.b	d0,(a2)
	lea.l	SPBANK(pc),a0		*仮スプライトデータエリア
	lea.l	$eb0000,a1		*本物のｽﾌﾟﾗｲﾄｽｸﾛｰﾙﾚｼﾞｽﾀ
	move.l	#255,d1
monitor1:
	move.l	(a0)+,(a1)+		*データ一気に書き込み
	dbra	d1,monitor1
	eor.b	d0,(a2)			*表示期間も待たずに表示
	rts
*
*==============================
SETENEMYSPB:
	move.w	#128,d7			*スプライトは128個だけ
	moveq.l	#31,d6			*敵は32匹だけ
	move.w	#$4000,d5		*ダメージビット判定用
	lea.l	ENEMYAREA(pc),a0
	lea.l	SPBANK(pc),a1		*仮ｽﾌﾟﾗｲﾄｽｸﾛｰﾙﾚｼﾞｽﾀ
STENMSPBLOOP:
	move.w	(a0)+,d4		*フラグ0なら空きエリア
	beq	SETENMNEXT

	move.w	(a0)+,d1		*計算用X座標
	addq.w	#2,a0
	move.w	(a0)+,d2		*計算用Y座標
	move.l	14(a0),a2		*パターンポインタヘッド
	movea.w	24(a0),a3		*プライオリティ
	move.w	18(a0),d0		*表示パターン
	add.w	d0,d0
	adda.w	0(a2,d0.w),a2		*やっとポインタからGデータ
	move.w	(a2)+,d0		*使用スプライト数
	beq	SETENMNEXT2		*0だったらなにも表示しない
	sub.w	d0,d7			*D7=残りスプライト数
	bcs	SETENEMYPUNK		*オーバーならこの敵は無視
	subq.w	#1,d0			*d0=カウンター
	and.w	d5,d4			*ダメージビット=0ならLOOP2
	beq	STENMSPBLP2
	and.w	#$bfff,-8(a0)		*敵エリアのﾀﾞﾒｰｼﾞﾋﾞｯﾄ消す
STENMSPBLP1:				*---------damaged!!--------
	move.w	(a2)+,d3		*オフセットX座標
	add.w	d1,d3
	move.w	d3,(a1)+
	move.w	(a2)+,d3		*オフセットY座標
	add.w	d2,d3
	move.w	d3,(a1)+
	move.w	(a2)+,d4		*V/H, Color, SP-CODE
	or.w	#$0f00,d4		*ダメージはカラー15
	move.w	d4,(a1)+
	move.w	a3,(a1)+		*プライオリティ
	dbra	d0,STENMSPBLP1		*使用数分だけループ
	bra	SETENMNEXT2
STENMSPBLP2:				*---------normal-----------
	move.w	(a2)+,d3		*オフセットX座標
	add.w	d1,d3
	move.w	d3,(a1)+
	move.w	(a2)+,d3		*オフセットY座標
	add.w	d2,d3
	move.w	d3,(a1)+
	move.w	(a2)+,(a1)+		*V/H, Color, SP-CODE
	move.w	a3,(a1)+		*プライオリティ
	dbra	d0,STENMSPBLP2		*使用数分だけループ
SETENMNEXT2:
	lea.l	64-8(a0),a0		*次の敵データのヘッド
	dbra	d6,STENMSPBLOOP		*残り敵数分ループ
	bra	SETENMEXIT
SETENMNEXT:
	lea.l	64-2(a0),a0		*次の敵データのヘッド
	dbra	d6,STENMSPBLOOP		*残り敵数分ループ
SETENMEXIT:
	subq.w	#1,d7			*仮レジスタの残り無し?
	bcs	SETENMEND		*D7=カウンタになります
	lea.l	6(a1),a1		*ﾚｼﾞｽﾀのプライオリティ部
	moveq.l	#0,d1			*非表示
SETENMEXIT2:
	move.w	d1,(a1)
	lea.l	8(a1),a1
	dbra	d7,SETENMEXIT2		*残りは全て非表示
SETENMEND:
	rts
*
SETENEMYPUNK:
	add.w	d0,d7			*残りスプライト数を戻す
	bra	SETENMEXIT		*次の敵へ
*
*============================================================
ENEMYMOVE:
	lea.l	ENEMYAREA(pc),a6
	moveq.l	#31,d7			*敵32匹
ENEMYMOVELOOP:
	move.w	(a6),d0			*敵フラグ
	beq	NEXTENEMYMOVE		*0なら空きエリア
	move.w	DAM(a6),d0		*ダメージならフラグ立てる
	beq	nodamage
	moveq.l	#0,d1			*エリアのダメージリセット
	move.w	d1,DAM(a6)
	move.b	d0,d1			*ダメージ最高255です
	sub.w	d1,VITAL(a6)		*耐久力減らす
	bcc	damaged			*まだ生きてるならそのまま
	move.l	DADR(a6),a4		*死亡ｱﾄﾞﾚｽと移動ｱﾄﾞﾚｽ交換
	move.l	MADR(a6),DADR(a6)
	move.l	a4,MADR(a6)
	bra	damagedend
damaged:
	or.w	#$4000,(a6)		*ダメージフラグ立てる
	moveq.l	#0,d0			*死んではないからd0=0
nodamage:
	move.l	MADR(a6),a4		*移動アドレス
damagedend:
	jsr	(a4)			*移動プログラムへ!
*
NEXTENEMYMOVE:
	lea.l	64(a6),a6		*次の敵
	dbra	d7,ENEMYMOVELOOP
	rts
*
*==================================================================
*==================================================================
*
PALET:					*パレット14、15
	dc.w	$0000,$01c0,$0280,$0380,$04c0,$0640,$4f40,$7f80
	dc.w	$9780,$bfca,$efd4,$ffde,$ffea,$fff2,$fffe,$003e
	dc.w	$0000,$fffe,$e738,$d6b4,$c630,$bdee,$ad6a,$9ce6
	dc.w	$94a4,$8420,$739c,$6318,$5294,$4a52,$39ce,$fffe
*
crtfst:					*起動時の画面モード
	dc.l	0
STICKWAY:				*ｼﾞｮｲｽﾃｨｯｸ入力のX, Y加算分
	dc.w	0,0,0,-3,0,3,0,0,-3,0,-2,-2,-2,2,-3,0
	dc.w	3,0,2,-2,2,2,3,0,0,0,0,-3,0,3,0,0
KEYWAY:					*カーソルからｼﾞｮｲｽﾃｨｯｸデータを
	dc.b	0,4,1,5,8,0,9,1,2,6,0,4,10,2,8,0
*--------------------------
ENEMYAREA:				*敵1個32ワード
*sample	:Flag.w:X.w:Y.w:XL.w:YL.w:Move.l:Dead.l:G-adr.l:Pat.w:Vitl.w:Damage.w:Prw.w:
	dc.w	1,1024+100,0,1024+100,0,16,16
	dc.l	ene1,ene1,EPAT1
	dc.w	0,100,0,3
	dc.w	0,0,0,0,0,0,0,0,0,0,0,0,0,0
enemy2:
	dc.w	1,1024+100,0,1024+100,0,16,16
	dc.l	ene2,ene2,EPAT1
	dc.w	0,100,0,3
```

```
	dc.w	16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
enemy3:
	ds.w	32*30
*
*----------------------------
SPBANK:		ds.w	4*128			*仮ｽﾌﾟﾗｲﾄｽｸﾛｰﾙﾚｼﾞｽﾀ
*
XYAREA:						*オプション用XYスタック
	ds.w	2*1024
XYAREALAST:
*
*==========================================
ene1:						*敵頭のプログラム
	moveq.l	#7,d1
	IOCS	_BITSNS		*カーソルキー判定
	lsr.w	#3,d0
	and.w	#$0f,d0
	lea.l	KEYWAY(pc),a0	*カーソルから方向を得る
	move.b	0(a0,d0.w),d1
	move.b	$e9a001,d0	*ｽﾃｨｯｸ1
	not.w	d0		*1で入力にする
	or.b	d1,d0		*キーボードと合成
	and.w	#$000f,d0	*方向のみ要る
	lsl.w	#2,d0		**4
	lea.l	STICKWAY(pc),a0	*入力によりX，Y取る
	move.w	0(a0,d0.w),d1
	move.w	2(a0,d0.w),d2
	move.w	d1,d0
	or.w	d2,d0
	beq	ene1ok		*動いてなければ飛ばす
	add.w	ENEX(a6),d1
	add.w	ENEY(a6),d2
	move.w	#1024+1,d3	*X，Yクリッピング
	move.w	#1024+256-31,d4
	cmp.w	d3,d1
	bcc	enex1
	move.w	d3,d1
enex1:
	cmp.w	d4,d1
	bcs	enex2
	move.w	d4,d1
enex2:
	cmp.w	d3,d2
	bcc	eney1
	move.w	d3,d2
eney1:
	cmp.w	d4,d2
	bcs	eney2
	move.w	d4,d2
eney2:
	move.w	d1,ENEX(a6)	*新しいX，Y
	move.w	d2,ENEY(a6)

	lea.l	XYAREALAST-4(pc),a0
	lea.l	XYAREALAST(pc),a1
	move.w	#1024-2,d0
ene1loop:				*オプション用XY操作
	move.l	-(a0),-(a1)
	dbra	d0,ene1loop
	move.w	d1,(a0)+
	move.w	d2,(a0)+
ene1ok:
	rts
*
ene2:					*オプションのプログラム
	move.w	PAREA(a6),d2	*頭からのデータ上の距離
	add.w	d2,d2		*XY1組だから
	lea.l	XYAREA(pc),a0
	move.w	0(a0,d2.w),d0	*X
	move.w	2(a0,d2.w),d1	*Y
	move.w	d0,ENEX(a6)
	move.w	d1,ENEY(a6)
	cmp.w	#1024+128,d0	*ダメージ判定
	bcs	ene2not
	cmp.w	#1024+128+32,d0
	bcc	ene2not
	cmp.w	#1024+128,d1
	bcs	ene2not
	cmp.w	#1024+128+32,d1
	bcc	ene2not
	bset.b	#6,(a6)		*食らったらフラグ立てる
ene2not:
	move.w	PAREA+2(a6),d0	*アニメーションカウンタ
	addq.w	#1,d0
	and.w	#15,d0
	move.w	d0,PAREA+2(a6)
	lsr.w	#2,d0
	move.w	d0,PAT(a6)	*カウンタからパターンへ
	rts
*
EPAT1:					*アニメーション用ポインタ
	dc.w	egr0-EPAT1,egr1-EPAT1,egr2-EPAT1,egr1-EPAT1
egr0:		dc.w	4,15,15,$0e40,31,15,$4e40,15,31,$8e40,31,31,$ce40
egr1:		dc.w	4,15,15,$0e41,31,15,$4e41,15,31,$8e41,31,31,$ce41
egr2:		dc.w	4,15,15,$0e42,31,15,$4e42,15,31,$8e42,31,31,$ce42
*
*========================================================
SETSPRITE:
	lea.l	$eb8000+128*$40,a1	*コード$40
	lea.l	OPTION(pc),a0
	bsr	WRITE128		*$40
	bsr	WRITE128		*$41
	bsr	WRITE128		*$42
	rts
*
WRITE128:
	moveq.l	#31,d7
WRITE128a:
	move.l	(a0)+,(a1)+
	dbra	d7,WRITE128a
	rts
*
OPTION:				*いい加減なスプライトパターン
	dc.w	$0000,$0000,$0000,$0000,$0000,$0000,$0000,$0011
	dc.w	$0000,$0011,$0000,$1122,$0000,$1122,$0011,$2233
	dc.w	$0011,$2233,$0022,$3344,$0022,$3344,$2233,$5555
	dc.w	$2233,$5555,$2233,$5577,$2233,$5577,$2233,$5577
	dc.w	$0000,$0000,$0011,$1111,$0011,$1111,$1133,$3333
	dc.w	$1133,$3333,$3344,$4444,$3344,$4444,$4455,$6666
	dc.w	$4455,$6666,$6666,$8888,$6666,$8888,$7788,$9999
	dc.w	$7788,$9999,$9999,$aabb,$9999,$aabb,$99aa,$eeee
*2
	dc.w	$0000,$0000,$0000,$0000,$0000,$0000,$0000,$0000
	dc.w	$0000,$0000,$0000,$0000,$0000,$0000,$0000,$0011
	dc.w	$0000,$0011,$0000,$1122,$0000,$2233,$0000,$2233
	dc.w	$0002,$3355,$0002,$3355,$0002,$3355,$0002,$3355
	dc.w	$0000,$0000,$0000,$0000,$0000,$0000,$0001,$1111
	dc.w	$0001,$1111,$1113,$3333,$1113,$3333,$2331,$4444
	dc.w	$2334,$4444,$3445,$5666,$4666,$6888,$4666,$6888
	dc.w	$5778,$8999,$5778,$8999,$7999,$9aab,$7999,$9aab
*3
	dc.w	$0000,$0000,$0000,$0000,$0000,$0000,$0000,$0000
	dc.w	$0000,$0000,$0000,$0000,$0000,$0000,$0000,$0000
	dc.w	$0000,$0000,$0000,$0000,$0000,$0001,$0000,$0001
	dc.w	$0000,$0002,$0000,$0223,$0000,$0223,$0000,$0223
	dc.w	$0000,$0000,$0000,$0000,$0000,$0000,$0000,$0000
	dc.w	$0000,$0000,$0000,$0000,$0000,$1111,$0000,$1111
	dc.w	$0111,$3333,$1223,$4444,$2334,$5666,$2331,$5666
	dc.w	$3446,$6888,$5557,$8999,$5779,$9aab,$5779,$9aab
*
	.end
```

リスト2

```
*
*================================
SETENEMYSPB:
		move.w	#128,d7		*スプライトは128個だけ
		moveq.l	#31,d6		*敵は32匹だけ
		move.w	#$4000,d5	*ダメージビット判定用
		lea.l	ENEMYAREA(pc),a0
		move.l	BANKHEAD,a1
		lea.l	SPBANK+8*128(pc),a4
		lea.l	8*32(a1),a1
		cmp.l	a4,a1
		bcs	STENMSPB1
		lea.l	SPBANK(pc),a1
STENMSPB1:
		move.l	a1,BANKHEAD
STENMSPBLOOP:
		move.w	(a0)+,d4	*フラグ0なら空きエリア
		beq	SETENMNEXT

		move.w	(a0)+,d1	*計算用X座標
		addq.w	#2,a0
		move.w	(a0)+,d2	*計算用Y座標
		move.l	14(a0),a2	*パターンポインタヘッド
		movea.w	24(a0),a3	*プライオリティ
		move.w	18(a0),d0	*表示パターン
		add.w	d0,d0
		adda.w	0(a2,d0.w),a2	*やっとポインタからGデータ
		move.w	(a2)+,d0	*使用スプライト数
		beq	SETENMNEXT2	*0だったらなにも表示しない
		sub.w	d0,d7		*D7=残りスプライト数
		bcs	SETENEMYPUNK	*オーバーならこの敵は無視
		subq.w	#1,d0		*d0=カウンター
		and.w	d5,d4		*ダメージビット=0ならLOOP2
		beq	STENMSPBLP2
		and.w	#$bfff,-8(a0)	*敵エリアのﾀﾞﾒｰｼﾞﾋﾞｯﾄ消す
STENMSPBLP1:				*---------damaged!!---------
		move.w	(a2)+,d3	*オフセットX座標
		add.w	d1,d3
		move.w	d3,(a1)+
		move.w	(a2)+,d3	*オフセットY座標
		add.w	d2,d3
		move.w	d3,(a1)+
		move.w	(a2)+,d4	*V/H, Color, SP-CODE
		or.w	#$0f00,d4	*ダメージはカラー15
		move.w	d4,(a1)+
		move.w	a3,(a1)+	*プライオリティ
		cmp.l	a4,a1
		bcs	OVER1
		lea.l	SPBANK(pc),a1
OVER1:
		dbra	d0,STENMSPBLP1	*使用数分だけループ
		bra	SETENMNEXT2
STENMSPBLP2:				*---------normal------------
		move.w	(a2)+,d3	*オフセットX座標
		add.w	d1,d3
		move.w	d3,(a1)+
		move.w	(a2)+,d3	*オフセットY座標
		add.w	d2,d3
		move.w	d3,(a1)+
		move.w	(a2)+,(a1)+	*V/H, Color, SP-CODE
		move.w	a3,(a1)+	*プライオリティ
		cmp.l	a4,a1
		bcs	OVER2
		lea.l	SPBANK(pc),a1
OVER2:
		dbra	d0,STENMSPBLP2	*使用数分だけループ
SETENMNEXT2:
		lea.l	64-8(a0),a0	*次の敵データのヘッド
		dbra	d6,STENMSPBLOOP	*残り敵数分ループ
		bra	SETENMEXIT
SETENMNEXT:
		lea.l	64-2(a0),a0	*次の敵データのヘッド
		dbra	d6,STENMSPBLOOP	*残り敵数分ループ
SETENMEXIT:
		subq.w	#1,d7		*仮レジスタの残り無し?
		bcs	SETENMEND	*D7=カウンタになります
		lea.l	6(a1),a1	*ﾚｼﾞｽﾀのプライオリティ部
		moveq.l	#0,d1		*非表示
SETENMEXIT2:
		move.w	d1,(a1)
		lea.l	8(a1),a1
		cmp.l	a4,a1
		bcs	OVER3
		lea.l	SPBANK+6(pc),a1
OVER3:
		dbra	d7,SETENMEXIT2	*残りは全て非表示
SETENMEND:
		rts
*
SETENEMYPUNK:
		add.w	d0,d7		*残りスプライト数を戻す
		bra	SETENMEXIT	*次の敵へ
*
BANKHEAD:
		dc.l	SPBANK
```

BEMSによるキャラクタ管理

Kageyama Hiroaki 影山 裕昭

リアルタイムゲーム中でのキャラクタ管理のひとつのサンプルとして，BEMSを紹介します。基本的を考え方はわかりやすいので，これを叩き台としてキャラクタ管理の勘所を探ってみるとよいでしょう。

BEMS復活！

こんにちは影山です。いきなりで恐縮ですけど，いま一度X68000を買った理由を思い出してみてください。なに，アーケードからの移植ソフトがたくさんあるから？ そういう輩はいますぐスーパーファミコンを買いに行きなさい。65536色でグラフィックを描きたい，MIDIで作曲活動をしたい，スプライトバシバシのゲームを作りたい，そんな熱い気持ちを抱いてX68000を手にした人がほとんどだと思います。

このなかでも特に「ゲームを作りたかった人たち」というのは，ゲームで遊ぶことは多くとも，自分でもゲームを作ってますっていえる人が少ないようです。まぁ，だいたいの理由は想像がつきます。絵を描きたければ，Z's STAFFやMATIERを買ってくればいい。MIDIをやりたければ，MIDI楽器とZ-MUSICに代表されるMIDI/FM/AD PCM音源ドライバを揃えればいい。ゲームを作りたい人は……スプライトエディタでキャラクタパターンを作るくらいはできるでしょうが，これといった道具（ツール)がありません。シューティング68Kくらいのものでしょうか。しかしシューティング68Kでも，作成できるゲームはどれもサンプルのアレンジ版のようなものにしかなりません。もっと自由度がほしいところです。

誰でも一度くらいはゲームを遊んでいて，「こんなの作ってみたいなぁ」と漠然と感じたことがあると思います。それはたとえば，ドラクエのようなゲームだったり，ジェノサイドのようなゲームだったりするんでしょう。でも，なにをどうすればいいかわからない。スプライトの移動や表示は？ スプライトの衝突判定は？ いきなりやろうたってできなくて当たり前。すべてを一から作り出すのは大変な作業です。

昔からOh!Xでは読者のゲームプログラム作成を支援するかたちのツールを提供してきました。最近ではCARD.FNC, MAGIC4.Xなど。CARD.FNCはそこそこ投稿があってOh!X誌面でも何回か発表されていますが，その他のツールを使用した投稿作品は非常に少ないようです。これは以前のOh!Xのゲーム特集で中野氏がいっていたことの繰り返しになりますが，プログラムの技術的問題以前に根性がないということになるのでしょう。SION IIを制作する浜崎氏の姿を見れば，それもうなずけます。

前述したように，X68000にはMAGIC4.X, CARD.FNCといったゲーム作成支援ツールが発表されているけども，スプライトを使ったゲーム作成を支援するツールは発表されていませんでした。それを知ってか，今回のゲーム特集に際して編集者から「BEMSをX68000に移植してみない？」と声をかけられたのです。

BEMSとはなにか

本誌がOh!MZと呼ばれていた頃からの読者は知ってるでしょうが，BEMSとは1985年8月号で発表されたS-OS上でのリアルタイムゲーム作成を支援するゲーム開発パッケージのことです。原作はガイガーカウンタの制作，inside X68000の執筆などでOh!X読者にはお馴染みの桒野氏。BEMSはリアルタイムゲームが自機（Ship）と敵(Enemy)とミサイル(Missile)と背景(Back)の4つの属性で構成されているという仮定を踏まえて制作されています（これら4つの属性の頭文字をとってBEMSと名づけられました）。

リアルタイムゲームの代表であるシューティングゲーム（グラディウスやファンタジーゾーン）の大まかなゲーム構成を考えてみましょう。自機(Ship)と，敵(Enemy)がいて，お互いにレーザーやミサイル(Missile)を発射して相手を攻撃します。また背景(Back)には，宇宙空間があったり，パステルカラー調の風景があったりします。まさにB，E，M，Sの4つの属性で構成されているではありませんか！ バブルボブルやパックマンなどのアクションゲームなども突き詰めていくとBEMSの要素に分解できます。

実際のBEMSを使ったゲーム作りは，これら4つの属性ごとにキャラクタの移動ベクトル，移動スピード，さらに異なる(もしくは同じ)属性同士で衝突判定を行うかどうかといった情報を，BEMSの持っている4つのテーブルに設定します。あとはメインルーチン側でBEMSを呼び出すだけで，BEMSがキャラクタの移動，表示，衝突判定といった面倒な処理をこなしてくれるのです。これによってプログラマはこれらの面倒な処理を考えることなく，プログラミングに集中することができるのです。あぁ，なんて素晴らしいシステムなんでしょう。

ですが結果的にBEMSのバージョンアップがされなかったことからわかるように，BEMSは話題にはなりましたが，BEMS自体を使った作品にはあまりお目にかかれませんでした。概念的にわかりにくく，やはり他人の作ったツールを使い込むには相当の時間がかかるようで，多くの読者がBEMSを使いこなすまでいかなかったようです。

最近ではMAGIC4.Xについても同じことがいえるのではないでしょうか。MAGICを使ったサンプルゲーム，SION→SION IIの変化は目を見張るものがありました。SION IIをサンプルと呼ぶには抵抗を感じる人も多いでしょう。いま動いているSION III(仮)は，座標系がひとつしかない MAGICで，「よくぞここまで！」というような動きをしています。作者の浜崎氏は本当に頑張っています。おそらく日本でいちばん MAGICを使いこなしてる人物でしょう。

今回提供する開発ツールは，基本的に原作のBEMSと同じ考えで作られています。しかしS-OS専用からX68000専用になったということで，キャラクタがテキストからスプライトに変更されていたり，複数のスプライトを組み合わせた大きなパターンも扱えるようになっています。読者の皆さんもSION IIに感動しているだけでなく，自分でもMAGIC4.XやBEMSを使ったゲーム作品を制作して，バシバシOh!Xに投稿してきてください！

BEMSの概要

BEMSはリアルタイムゲームのなかで，スプライトの移動，表示，衝突判定処理を引き受けるゲーム作成支援ツールであることはすでに話しました。BEMSではBをBG0，E，M，Sをスプライトで表示します。現バージョンではBは表示のみで移動することができません。またBはメインプログラム側でBGを使用することを指定しないと，BEMSで処理することはできません。このような仕様になってるのは，場合によってはBGを使わずひとつでも多くのPCGデータエリアを確保したいことがあるのではと考えたからです。

ここでBEMSのキャラクタ管理の仕様について触れておきましょう。

1)　BEMSで扱うキャラクタはひとつのスプライト，もしくは複数のスプライトから構成されます。BEMSでは複数のスプライトをパターンとして扱えるので，たとえばキャラクタの大きさを横4×縦4に設定すれば，疑似的に64×64のスプライトとして表示することができます。

以後，説明中のキャラクタとはこの定義に従ったものです。

2)　X68000は最大同時表示スプライト数の128個を超えたスプライトを表示することはできませんが，B，E，M，Sのそれぞれのキャラクタについて，最大65535個まで登録することができます（無意味？）。

3)　キャラクタの表示座標はX68000のスプライト画面の座標に準じます。したがって表示画面が512×512では画面左上座標が(16，16)，画面右下座標が(528，528)となります。

4)　キャラクタの移動方向はX方向，Y方向のベクトルの大きさで指定します。

5)　キャラクタの移動速度は128段階で調節できます。0を設定した場合は，スプライトの移動を停止します。

6)　キャラクタごとに移動ドット単位を指定することができます。

7)　衝突判定はB，E，M，Sの各属性間で行うか否かを選択できます。たとえば，BとEは衝突判定をしますが，BとMは衝突判定をしない，といったことができます。

8)　各キャラクタごとに衝突判定を行う範囲を設定することができます。見た目に自然な衝突判定になります。

9)　スプライトコントロールレジスタの持っている機能である水平・垂直反転，表示プライオリティの設定ができます。

＊　＊　＊

このように，BEMSを使用することでプログラム作成の際の手間や考え方が非常に楽になります。X68000のスプライトは16×16ドットの大きさであることを意識せずにゲームデザインができ，衝突判定も自動的に行われます。スプライトの衝突判定では，以前「ショートプロぱーてぃ」でSP_CHK()というX-BASICの外部関数が提供され，おおいに重宝されていましたが，このBEMSでは当たり範囲の設定などもできるのでさらにプログラムの負担が軽減されると思います。

BEMSでのゲーム作成

BEMSはキャラクタにB（背景），E（敵），M（ミサイル），S（自機）の4つの属性を割り当て，各属性をテーブルによって管理します。キャラクタ移動に関してはE，M，Sは内部的にまったく同じ処理です。まず，E，M，Sの属性を持つキャラクタの管理方法を説明しましょう。

E，M，Sの各キャラクタは，32バイトのベクトルテーブルによって管理されます。ベクトルテーブルには，各キャラクタの表示座標，移動方向，キャラクタのパターンサイズ，衝突判定を行う範囲，表示するスプライト番号とスプライトパターン番号の対応が定義されているアドレスなどを格納します。同じ属性で複数のキャラクタを表示する場合は，連続したメモリ領域にベクトルテーブルを定義します。

Bを使う場合はユーザー自身がBG0(BG1)にパターンを表示します。そのときBとE，M，Sのいずれかの間で衝突判定をする場合は，BGに表示するパターンコードに約束事があります。今回発表するBEMSでは，PCGコードのパターン番号をある番号で分断して，分断したパターン番号以上を衝突判定をするB，未満を衝突判定をしないBとしています。たとえば64で分断すれば，パターン番号0～63は衝突判定のないB，パターン番号64以上は衝突判定のあるBということです。分断番号をいくつにするかは，ソースリストのアセンブルの際に指定します。ということなので，BのテーブルはBG0（BG1）の置かれているアドレス領域そのものです。

これらのベクトルテーブルがどのアドレスにあり，定義されているキャラクタがいくつあるか示すのがDEFPTR(Definition pointer)テーブルです。ただしBについては，テーブルの開始アドレスをBG0なら$EBC000，BG1なら$EBE000をユーザーが指定しなければなりません。また長さは固定になってますので指定しても意味を持ちません。

これらベクトルテーブルとDEFPTRテーブルによって，BEMSはキャラクタ移動を管理しています。次にBEMSの衝突管理について説明しましょう。

BEMSでは各属性間で衝突判定をするかしないかをCOLCTL(Collision control)テーブルに指定します。衝突が発生した場合，BEMSは衝突発生のフラグを立て，2つのキャラクタの属性，ベクトルテーブルの先頭アドレスをSDTBL(Collison source & destination table)に格納してメインルーチンに処理を戻します。

メインルーチン側ではBEMSを呼び出したあとにSDTBLの衝突発生のフラグを調べ，衝突が発生してなければ処理を先に進めるようにします。衝突が発生したのなら，衝突した2つの属性がSDTBLに格納されてるのでこれを調べます。たとえばMとEが衝突したのなら，自機の発射したミサイルが敵に当たったと判断できますので，敵とミサイルを消去してスコアをカウントアップといった処理をします。キャラクタの消去はベクトルテーブルを1バイト書き換えるだけで，とても簡単です。

なお，衝突発生後にBEMSを呼び出すと，BEMSは衝突発生直後から継続してキャラクタの移動や衝突判定処理を行います。

BEMSの4つのテーブル解説

これまでの説明でおわかりのように，BEMSはすべてのキャラクタ管理をテーブルで行っています。BEMSを使いこなすには，これら4つのテーブルの構造をしっかりと理解することにほかなりません。それではBEMSの各テーブルの構成を説明しましょう。

1)　ベクトルテーブル

E，M，Sの属性のキャラクタの移動に

関するパラメータを格納するテーブルです。ベクトルテーブルの大きさはひとつのキャラクタにつき32バイトで，その内容は図1に示すとおりです。テーブルの先頭から順番に説明しましょう。なお，名称の先頭に＊をつけたものは，そこが書き込み禁止であることを示します（書き込んでも動作の保証はありません）。

●ENBL_CTRL(Display enable control)

ここを1にするとキャラクタを表示，0にするとキャラクタを非表示にします。0以外ならなにを書いても表示になりますが，桑野氏の当時の原稿にも表示は1，非表示は0とあるので，それを守ることにしましょう。

●＊VECT_BUF

BEMSが移動の計算を行うためのワークエリア。ここは書き換えないでください。

●POS_X，POS_Y

キャラクタを表示する座標を格納します。スプライト座標なので，表示画面の左上座標が(16，16)になることに注意してください。ここの値はBEMSの移動計算によって逐次変更されます。

●VECT_X，VECT_Y

キャラクタの移動方向を決定します。VECT_Xが画面横方向，VECT_Yが画面縦方向の移動量を示します。VEXT_Xは右方向，VECT_Yは下方向に進むのがプラスで，反対方向がマイナスになります。マイナスの指定は2の補数表現で行います。キャラクタの移動方向はVECT_XとVECT_Yの比で決定されます。右上45度に移動す

VECT_X＝1
VECT_Y＝－1

となります。左下に緩やかに移動させるなら，ら，

VECT_X＝－8
VECT_Y＝1

という具合になるでしょうか。またVECT_X，VECT_Yは移動方向を決定するだけなので，

VECT_X＝3
VECT_Y＝5

と，

VECT_X＝6
VECT_Y＝10

はまったく同じ動きになります。

●＊SPEED_BUF

キャラクタ移動の計算に必要なワークエリアです。書き換えないでください。

●SPPED

キャラクタの移動速度を指定します。127が最高速で0で停止します。数値に比例した移動速度になるので，SPEEDが2と4のキャラクタでは移動時間に2倍の差があります。

●MOVE_DOT

キャラクタの移動単位を指定します。ここにnを書き込むとnドット単位でキャラクタを移動させます。

●PAD

ダミーです。

●CHRSIZE_X，CHRSIZE_Y

キャラクタの大きさをスプライト単位で指定します。CHRSIZE_Xが横方向に表示するスプライトの個数，CHRSIZE_Yが縦方向に表示するスプライトの個数を示します。

大きなパターンを表示する場合，たとえば6個のスプライトを使って横3×縦2で48×32のパターンとして定義するには，

CHRSIZE_X＝3
CHRSIZE_Y＝2

とします。画面には図2のような順番でスプライトを定義順に表示します。なお表示するスプライト番号とパターン番号の定義は，後述のPATTERNに格納されているアドレスからの内容に従います。

●HITCHK_X1，HITCHK_Y1，HITCHK X2，HITCHK_Y2

キャラクタの衝突判定を行う範囲を長方形の領域で指定します。範囲の指定方法は図3のようにキャラクタの左上を(0，0)として，衝突判定範囲の左上点を(HITCHK_X1，HITCHK_Y1)に，右下点を(HITCHK_X2，HITCHK_Y2)に指定します。必ずHITCHK_X1<HITCHK_X2，HITCHK_Y1<HITCHK_Y2となるように指定してください。

●VR_HR_COLOR

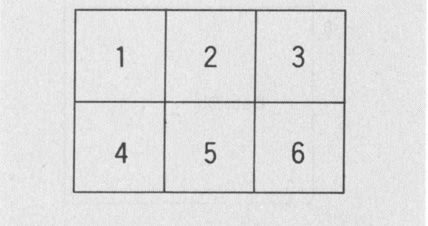
キャラクタの水平/垂直反転，使用するパレットブロックを指定します。各機能の指定方法を図4に示します。

●PRW

スプライトとBGとの表示優先順位の関係を設定します。指定できる数値は0～3です。以下にスプライトとBGの表示関係を示します。

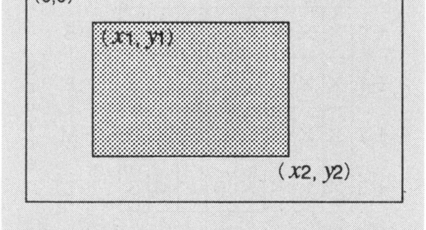
0．．スプライトを表示しません

図2　複数のスプライトで大きなキャラクタを表示する場合



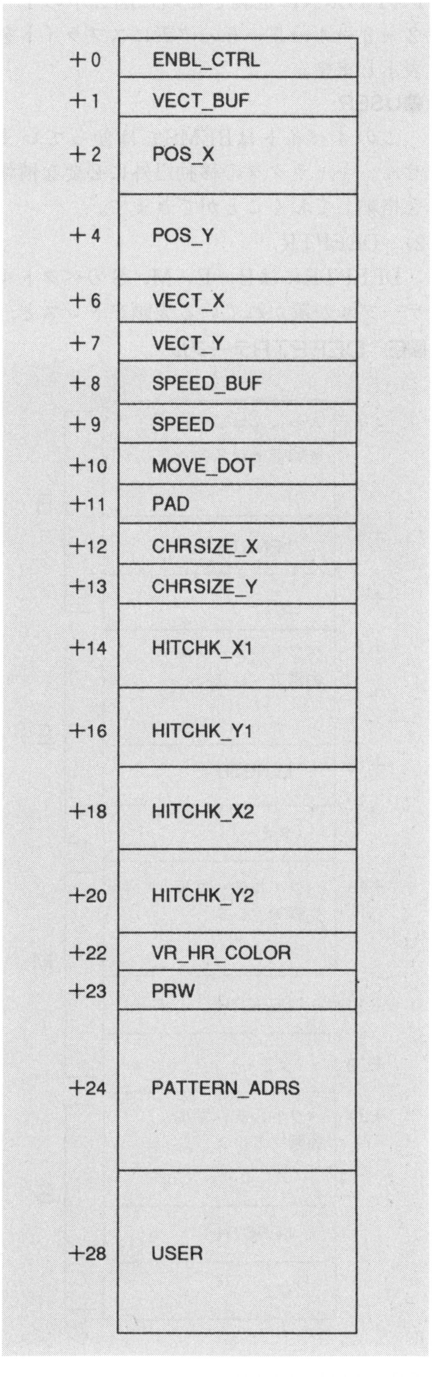
図中の番号順にスプライトデータを表示する

図3



図中の網点部分を衝突判定の対象範囲とする

図1　ベクトルテーブル



1 .. BG0>BG1>SP
2 .. BG0>SP>BG1
3 .. SP>BG0>BG1

●PATTERN_ADRS

スプライト番号に定義するPCGパターン番号を記述した領域の先頭アドレスを格納します。PATTERNで示されるアドレスには,

dc.b スプライト番号,PCGパターン番号
⋮
⋮

という形式でスプライトに定義するPCGパターン番号を指定します。複数のスプライトで大きなキャラクタを表示する場合は,PATTERNに定義した順に図2中の1→2→3→4→5→6の位置にスプライトを表示します。

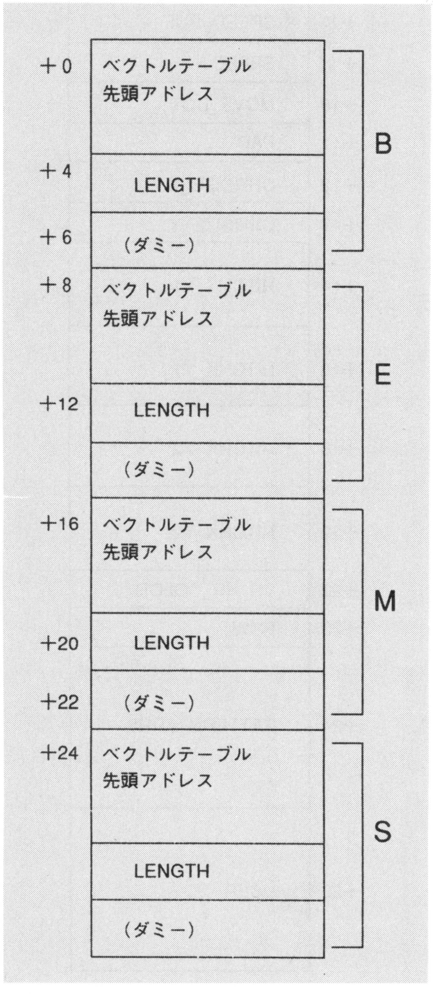
●USER

この4バイトはBEMSでは使っていません。キャラクタの移動以外に必要な情報を格納しておくことができます。

2) DEFPTR

DEFPTRにはB, E, M, Sのベクトルテーブルが置かれている先頭アドレスと,

図5 DEFPTRテーブル



定義されているキャラクタの個数を格納します。テーブルの構成は図5のようになっています。各属性には8バイトの要素があり,先頭の1ロングワード(ADRS)にベクトルテーブルの先頭アドレス,次の1ワード(LENGTH)にベクトルテーブルに定義してあるキャラクタ数を格納します。次の2バイトは使われていません。要素数が2のn乗だと内部での計算が楽になるので,このようにしてあります。

前にも簡単に説明しましたが,Bを使ってE, M, Sと衝突判定をする場合は,DEFPTRのBのベクトルテーブルにはBで使うBGの先頭アドレス(BG0なら$EBC000, BG1なら$EBE000)を格納してください。衝突判定を必要としない場合や,BGの表示やスクロールなどをメインルーチンで処理するときは,特にBのベクトルテーブルを設定する必要はありません。またBのLENGTHは固定なので,変更しても意味を持ちません。

3) COLCTL

B, E, M, Sの各属性の間で衝突の判定を行うかどうかを指定するテーブルです。COLCTLは図6に示すように,4バイトのテーブルとなっています。テーブルは横軸がデスティネーション(ぶつけられる側),縦軸がソース(ぶつかる側)になっています。衝突判定をするには1,しないなら0を指定します。

たとえばMがEにぶつかりにいったか衝突判定を行うには,COLCTL+2バイト目

図4 VR_HR_COLOR

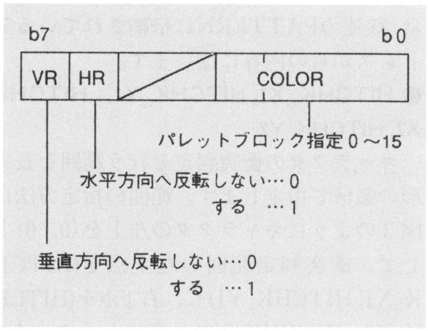
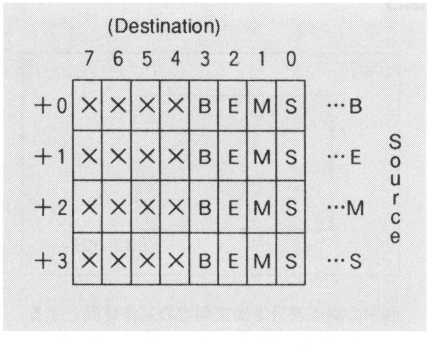


図6 COLCTLテーブル



の第2ビットを1にします。しかし,実際にはぶつけられる側とぶつかる側の区別はつけづらいので,対角線を境にして対称なビットは同じ値を入れておくのが普通でしょう。

対角線上に1を入れた場合は,同じ属性の間で衝突判定をすることになります。同じ属性同士で衝突判定をすると必ず自分自身との衝突が発生します。しかし,これは意味がないので自分自身との衝突判定は行わないようになっています。

またソースがBのところにもテーブルを持たせてありますが,これはかたちだけであって指定しても無視されます。

4) SDTBL

SDTBLの構成を図7に示します。BEMSでは衝突が発生したか否かの結果をSDTBLに格納します。衝突が発生すると,どのキャラクタとどのキャラクタが衝突したか,などの情報もSDTBLに格納されます。メインルーチン側ではSDTBLを参照して衝突発生後の処理をすることができます。

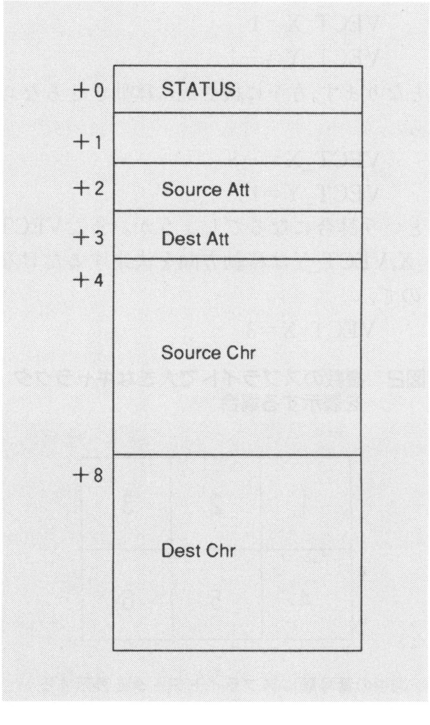
●STATUS

衝突が発生したかどうかを示します。衝突が発生したときは1,すべてのキャラクタについて移動,衝突処理を終えたのなら0が入ってます。

●Source Att, Dest Att

衝突したキャラクタと衝突されたキャラクタの属性が数値で入っています。Bは0,Eは1,Mは2,Sは3です。

図7 SDTBL



●Source Chr, Dest Chr

衝突したキャラクタと衝突されたキャラクタのベクトルテーブルのアドレスが入ります。衝突が発生した場合は，メインルーチン側でここを調べれば，具体的にどのキャラクタ同士が衝突したか知ることができます。

BEMSの作成

BEMSの作成方法を説明しましょう。まず，リスト１のソースリストをED.Xなどのエディタで入力します。すべて入力し終わったらアセンブルします。エラーが表示されたら，エディタで間違い箇所を訂正してください。BEMSはほかのプログラムとリンクして使われるものなので，このあとリンク作業を行う必要はありません（BEMS.Oのままでいい）。

普通にアセンブルすると，表示画面が256×256専用で使うことのできるBEMSが作成されます。またデフォルトではBGに表示されているパターン番号が120以上のものと，E，M，Sの間で衝突判定を行うようになっています。これを変更するには，BEMSをアセンブルするときに，シンボルを定義してください。まず表示画面512×512でBEMSを使いたい場合には，

A>AS /s512 BEMS.S

とシンボル512を定義してください。また衝突判定を行うパターン番号の下限を設定するには，HAS.Xなら，

A>AS /sBG_HIT_MAX=n BEMS.S

として，ｎに設定したい値を入れてください。AS.Xの場合はソース中にegu擬似命令で定義してください。

サンプルゲーム

説明だけではわからないところが多いと思いますので，BEMSを使ったサンプルプログラムを紹介します。このプログラムは自分（S）を襲ってくるゴキブリ（E）から逃げながら，ミサイル（M）を撃ってゴキブリを撃退するというゲームです。Ｂは使っていません。キャラクタはその場に居合わせた浜崎氏に強引に頼んで作成してもらいました（感謝）。

このプログラムは表示画面256×256で動作しますので，

A>AS BEMS

A>AS SAMPLE

とアセンブルしてから，

A>LK SAMPLE BEMS

としてメインルーチンとBEMSをひとつの実行ファイルにまとめます。

A>SAMPLE

でゲームが始まります。

APIC.Rなどで65536色の画像を表示してから実行すると，背景のグラフィックがスクロールします。テンキーの２，４，６，８で自機を動かすことができます。スペースキーでミサイルを発射します。

あくまでもサンプルなので，ゲーム性といったものはまったく考えていません(SIONシリーズはどうなんだ！ といわれると辛い)。プログラム内でのBEMSの使い方は特に解説しませんが，ソースリストがすべてを語っています。

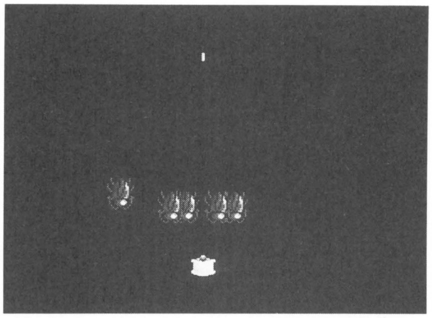
少し解説すると，サンプルプログラム(こちらがメインルーチン側）ではE，M，Sのベクトルテーブルを定義しておき，BEMS内部のテーブルやサブルーチンは外部参照ラベルとして定義しておきます。

BEMS.Oには４つのテーブルのうち，DEFPTR，COLCTL，SDTBLおよび，いくつかのサブルーチンを外部定義ラベルとして登録しています(９～15行)。

BEMS内部のサブルーチンの仕様を表１にまとめておきます。

最後に

今回発表するBEMSはBGのスクロールをサポートしていませんので，スクロールシューティングゲームの背景にBGを表示して自機と背景の衝突判定を行うということができません。私自身としては，BGのスクロール機能を付加したかったのですが，そのための時間がありませんでした。もしなんらかのかたちでBEMSのバージョンアップ版を発表できる機会があれば，BGスクロール機能を付加してみたいと思っています。



この原稿はOh!MZ1985年８月号の桑野氏の原稿を全面的に参考にして書かれています。桑野氏の解説は大変わかりやすく，移植も短期間に終えることができました。最後になりましたが，原作のBEMSを作成された桑野氏にこの場を借りて感謝の意を表します。

参考文献：

ゲーム開発パッケージBEMS，桑野雅彦，Oh!MZ 1985年８月号 44-51pp.

表１：

BEMS:
　キャラクタの移動，衝突判定を行います。処理はすべて４つのテーブルに従います。破壊レジスタはありません。
DISP:
　A1レジスタにベクトルテーブルの先頭アドレスを格納して呼び出すと，テーブルの内容に従ってキャラクタの表示，消去を行います。破壊レジスタはありません。
MOVE:
　A1レジスタにベクトルテーブルの先頭アドレスを格納して呼び出すと，テーブルの内容に従ってキャラクタの移動を行います。テーブルの内容は書き換えられます。破壊レジスタはありません。

リスト１ BEMS.S

```
 1: *
 2: * BEMS Original Program for S-OS Written by M.Kuwano 1984,1985
 3: *
 4: * BEMS X68000   Written by Shadow Mountain 1992
 5: *
 6:     .include        iocscall.mac
 7:     .include        bems.h
 8:
 9:     .xdef           BEMS
10:     .xdef           MOVE
11:     .xdef           DISP
12:     .xdef           COLL
13:     .xdef           DEFPTR
14:     .xdef           SDTBL
15:     .xdef           COLCTL
16:
17:     .text
18:     .even
19:
20: BEMS:
21:     movem.l d0-d7/a0-a6,-(sp)
22:     tst.b   SDTBL
23:     bne     CONT                        * 衝突直後の処理を再開
24:     move.b  #3,SDTBL+2
25: MAINLP:
26:     moveq.l #0,d7
27:     move.b  SDTBL+2(pc),d7
28:     lsl.b   #3,d7
29:     lea.l   DEFPTR(pc),a0
30:     movea.l (a0,d7.w),a1                * a1 = ﾍﾞｸﾄﾙﾃｰﾌﾞﾙの開始アドレス
31:     move.w  4(a0,d7.w),d6
32:     beq     NOTDSP                      * LENGTH = 0 なら
33:     subq.w  #1,d6
34: LOOP:
35:     bsr     MOVE
36:     bsr     COLL
37:     tst.b   SDTBL
38:     beq     CNTENT                      * 衝突は発生しなかった
39:     movem.l d0-d7/a0-a6,save_reg
40:     movem.l (sp)+,d0-d7/a0-a6
41:     rts
42: CNTENT:
43:     bsr     DISP
44: NOTDSP:
45:     lea.l   32(a1),a1
46:     dbra    d6,LOOP
```

```
	subq.b	#1,SDTBL+2
	bne	MAINLP
EXEEND:
	clr.b	SDTBL
	movem.l	(sp)+,d0-d7/a0-a6
	rts
*
* CONTINUE MODE
*
CONT:
	movem.l	save_reg(pc),d0-d7/a0-a6
	move.b	SPEED_BUF(a1),d7
	bsr	MOVE
	move.b	d7,SPEED_BUF(a1)
	tst.b	ENBL_CTRL(a1)
	beq	CNT1
	bsr	COLL
	bra	CNTENT
CNT1:
	bra	NOTDSP

*++++++++++++++++++++++++++++++++++++++++

DISP:
**
** in	ベクトルテーブルの先頭アドレス
**
	movem.l	d0-d7/a0-a6,-(sp)
	moveq.l	#0,d6
	moveq.l	#0,d7
	move.b	CHRSIZE_X(a1),d6		* 横方向に表示するスプライトの数
	move.b	CHRSIZE_Y(a1),d7		* 縦方向に表示するスプライトの数
	subq.w	#1,d6
	move.w	d6,d5
	subq.w	#1,d7
	movea.l	a1,a4
	lea.l	PATTERN_ADRS(a4),a4
	movea.l	(a4),a4			* a4 = ｽﾌﾟﾗｲﾄ番号格納アドレス
	moveq.l	#0,d2
	moveq.l	#0,d3
	move.b	VR_HR_COLOR(a1),d2
	lsl.w	#8,d2
	move.b	PRW(a1),d3
	lea.l	$eb0000,a3		* ｽﾌﾟﾗｲﾄｽｸﾛｰﾙﾚｼﾞｽﾀ先頭アドレス
	tst.b	ENBL_CTRL(a1)
	bne	DISP0
	moveq.l	#0,d3
DISP0:
	move.w	POS_Y(a1),d1		* パタン右上Y座標
DISP1:
	move.w	POS_X(a1),d0		* パタン左上X座標
DISP2:
	moveq.l	#0,d4
	move.b	(a4)+,d4		* ｽﾌﾟﾗｲﾄ番号
	lsl.w	#3,d4
	move.b	(a4)+,d2		* ﾊﾟﾀﾝ番号
*wait_vdisp:
*	btst.b	#4,$e88001
*	bne	wait_vdisp
	movem.w	d0-d3,(a3,d4.w)
	add.w	#16,d0			* POS_X=POS_X+16
	dbra	d6,DISP2
	move.w	d5,d6
	add.w	#16,d1			* POS_Y=POS_Y+16
	dbra	d7,DISP1
	movem.l	(sp)+,d0-d7/a0-a6
	rts

*++++++++++++++++++++++++++++++++++++++++

MOVE:
**
** in	ベクトルテーブルの先頭アドレス
**
	movem.l	d0-d7/a0-a6,-(sp)
	tst.b	ENBL_CTRL(a1)
	beq	MOVE_ret		* ENBL CTRL = 0なら移動、表示しない
	move.b	SPEED_BUF(a1),d7
	add.b	SPEED(a1),d7
	move.b	d7,SPEED_BUF(a1)	* SPEED_BUF=SPEED_BUF+SPEED
	subi.b	#127,d7
	bcs	MOVE_ret		* SPEED_BUFが127以下なら移動しない
	move.b	d7,SPEED_BUF(a1)

CALVCT:					* 移動ベクトルに基づいて次の表示座標を計算
	move.b	MOVE_DOT(a1),d4		* X方向の増加方向
	move.b	d4,d5			* Y方向の増加方向
	ext.w	d4
	ext.w	d5
VECT_X_check:
	move.b	VECT_X(a1),d6
	bpl	VECT_Y_check
	ext.w	d6
	neg.w	d6
	neg.w	d4			* X方向の増加方向マイナス
VECT_Y_check:
	move.b	VECT_Y(a1),d7
	bpl	VECT_check_end
	ext.w	d7
	neg.w	d7
	neg.w	d5			* Y方向の増加方向マイナス
VECT_check_end:
	tst.b	d6
	bne	skip
	tst.b	d7
	beq	MOVE_ret		* 移動ﾍﾞｸﾄﾙが0なら
skip:
	cmp.b	d6,d7
	bcc	CALVCT2			* dy>dxなら

	add.w	d4,POS_X(a1)
	bpl	CALVCT0
	add.w	#1024,POS_X(a1)
CALVCT0:
	andi.w	#$03ff,POS_X(a1)
	move.b	VECT_BUF(a1),d3
	add.b	d7,d3
	move.b	d3,VECT_BUF(a1)
	cmp.b	d6,d3
	bcs	MOVE_ret
	sub.b	d6,d3
	move.b	d3,VECT_BUF(a1)
	add.w	d5,POS_Y(a1)
	bpl	CALVCT1
	add.w	#1024,POS_Y(a1)
CALVCT1:
	andi.w	#$03ff,POS_Y(a1)
	bra	MOVE_ret

CALVCT2:
	add.w	d5,POS_Y(a1)
	bpl	CALVCT3
	add.w	#1024,POS_Y(a1)
CALVCT3:
	andi.w	#$03ff,POS_Y(a1)
	move.b	VECT_BUF(a1),d3
	add.b	d6,d3
	move.b	d3,VECT_BUF(a1)
	cmp.b	d7,d3
	bcs	MOVE_ret
	sub.b	d7,d3
	move.b	d3,VECT_BUF(a1)
	add.w	d4,POS_X(a1)
	bpl	CALVCT4
	add.w	#1024,POS_X(a1)
CALVCT4:
	andi.w	#$03ff,POS_X(a1)
MOVE_ret:
	movem.l	(sp)+,d0-d7/a0-a6
	rts

*++++++++++++++++++++++++++++++++++++++++

COLL:
	movem.l	d0-d7/a0-a6,-(sp)
	clr.b	SDTBL
	moveq.l	#0,d7
	move.b	SDTBL+2(pc),d7
	lea.l	COLCTL(pc),a0
TRIC2:
	move.b	(a0,d7.w),d6		* d6 = table buffer
	move.b	#3,SDTBL+3

	move.l	POS_X(a1),d5		* 上位16bit src POS_X,下位16bit src POS_Y
	move.l	HITCHK_X1(a1),d3	* 上位16bit src hit_chk_X1,下位16bit src h
it_chk_Y1
	move.l	HITCHK_X2(a1),d2	* 上位16bit src hit_chk_X2,下位16bit src h
it_chk_Y2
	add.l	d5,d3			* src X1,Y1座標
	add.l	d5,d2			* src X2,Y2座標
	lea.l	DEFPTR,a0
COLL1:
	lsr.b	#1,d6
	bcc	COLL2			* 衝突判定はしない
	moveq.l	#0,d5
	move.b	SDTBL+3(pc),d5
	lsl.b	#3,d5
	lea.l	(a0,d5.w),a2
	move.w	4(a2),d0		* d0 = Dest Chr LENGTH
	beq	COLL2
	subq.w	#1,d0
	movea.l	(a2),a2			* a2 = Dest Chr ﾍﾞｸﾄﾙﾃｰﾌﾞﾙ先頭ｱﾄﾞﾚｽ
	bsr	TBLSEC
	tst.b	SDTBL
	bne	COLEND
COLL2:
	subq.b	#1,SDTBL+3
	bne	COLL1
	roxr.b	#1,d6
	bcc	COLEND

* COLL WITH 'B' PLANE
*
* BGは動かさないものとする
*
	move.w	d3,d1			* Src Y1
	move.w	d2,d0			* Src Y2
	swap	d3			* Src X1
	swap	d2			* Src X2

	cmp.w	#16,d3
	bcc	skip1
	move.w	#16,d3
skip1:
	cmp.w	#16,d1
	bcc	skip2
	move.w	#16,d1
skip2:
	sub.w	#16,d1
	sub.w	#16,d3
	sub.w	#16,d0
	sub.w	#16,d2
COLLENT:
	sub.w	d3,d2
	sub.w	d1,d0

ifdef	512
	lsr.w	#4,d2			* dx-1
	lsr.w	#4,d0			* dy-1
	lsr.w	#4,d3			* Src X1/16
	lsr.w	#4,d1			* Src Y1/16
else
	lsr.w	#3,d2			* dx-1
	lsr.w	#3,d0			* dy-1
	lsr.w	#3,d3			* Src X1/8
	lsr.w	#3,d1			* Src Y1/8
endif
	move.l	DEFPTR,a2		* BG 先頭ｱﾄﾞﾚｽ
	movea.l	a2,a3
	add.w	d3,d3			* x*2
	adda.w	d3,a3
	lsl.l	#7,d1			* y*128
	adda.l	d1,a3			* BG_ADRS=BG_TOP_ADRS+y*128+x*2
	move.w	d2,a5
COLL_B0:
	movea.l	a3,a4
COLL_B1:

ifdef	BG_HIT_MAX
	cmpi.b	#BG_HIT_MAX,1(a4)
else
	cmpi.b	#120,1(a4)
endif
	bcc	COLL_B5
	addq.l	#2,a4
	dbra	d2,COLL_B1
	move.w	a5,d2
	lea.l	128(a3),a3
	dbra	d0,COLL_B0
	bra	COLEND
COLL_B5:
	cmpa.l	a1,a2
	beq	COLEND
	move.b	#1,SDTBL
	movem.l	a1/a4,SDTBL+4
COLEND:
	movem.l	(sp)+,d0-d7/a0-a6
	rts

TBLSEC:
	move.l	POS_X(a1),d5		* 上位16bit src POS_X,下位16bit src POS_Y
	move.l	HITCHK_X1(a1),d3	* 上位16bit src hit_chk_X1,下位16bit src h
it_chk_Y1
	move.l	HITCHK_X2(a1),d2	* 上位16bit src hit_chk_X2,下位16bit src h
it_chk_Y2
	add.l	d5,d3
	add.l	d5,d2
	move.l	POS_X(a2),d4		* 上位16bit dest POS_X,下位16bit dest POS_
Y
	move.l	HITCHK_X1(a2),d1	* 上位16bit dest hit_chk_X1,下位16bit dest
 hit_chk_Y1
	move.l	HITCHK_X2(a2),d7	* 上位16bit dest hit_chk_X2,下位16bit dest
```


▶恋に音楽はつきものだと思う今日この頃。皆さんは，何か曲を聞いているときに決まって特定の誰かを思い出すってことはありませんか？　小山　優一(18)東京都

```
 hit_chk_Y2
    add.l   d4,d1
    add.l   d4,d7

    cmp.w   d3,d7
    bcs     COLPSS                      * dest Y2<src Y1
    cmp.w   d2,d1
    bgt     COLPSS                      * dest Y1>src Y2
    swap    d1
    swap    d2
    swap    d3
    swap    d7
    cmp.w   d3,d7
    bcs     COLPSS                      * dest X2<src X1
    cmp.w   d2,d1
    ble     COLOCC                      * dest X1>src X2 (dest X1<=src X2 ナラ OK!
)
COLPSS:
    lea.l   32(a2),a2
    dbra    d0,TBLSEC
    rts
COLOCC:
    cmpa.l  a1,a2
    bne     COLRET
    rts                                 * 自分自身に衝突した場合をはじく
COLRET:
    tst.b   ENBL_CTRL(a1)
    beq     COLPSS                      * Src  Chr は表示しない
    tst.b   ENBL_CTRL(a2)
    beq     COLPSS                      * Dest Chr は表示しない
    move.b  #1,SDTBL
    movem.l a1/a2,SDTBL+4
    rts

    .data

DEFPTR:
    dc.l    $EBC000         * BG0 先頭アドレス
    ds.w    1               * LENGTH
    ds.w    1               * dummy
    ds.l    1               * E
    ds.w    1               * LENGTH
    ds.w    1               * dummy
    ds.l    1               * M
    ds.w    1               * LENGTH
    ds.w    1               * dummy
    ds.l    1               * S
    ds.w    1               * LENGTH
    ds.w    1               * dummy
SDTBL:
    ds.b    1               * STATUS
    ds.b    1               * PAD
    ds.b    1               * Source Att
    ds.b    1               * Dset   Att
    ds.l    1               * Source Chr Vector table address
    ds.l    1               * Dest   Chr Vector table address
COLCTL:
*               ----BEMS
    dc.b    %00000000       * B
    dc.b    %00000000       * E source
    dc.b    %00000000       * M
    dc.b    %00000000       * S.

save_reg:
    ds.l    15              * Save Register

    .end

```

リスト2 BEMS.H

```
* vector table

    .offset 0

ENBL_CTRL  ds.b    1
VECT_BUF   ds.b    1       * 1
POS_X              ds.w    1       * 2
POS_Y              ds.w    1       * 4
VECT_X             ds.b    1       * 6
VECT_Y             ds.b    1       * 7
SPEED_BUF  ds.b    1       * 8
SPEED              ds.b    1       * 9
MOVE_DOT   ds.b    1       * 10
PAD                ds.b    1       * 11
CHRSIZE_X  ds.b    1       * 12
CHRSIZE_Y  ds.b    1       * 13
HITCHK_X1  ds.w    1       * 14
HITCHK_Y1  ds.w    1       * 16
HITCHK_X2  ds.w    1       * 18
HITCHK_Y2  ds.w    1       * 20
VR_HR_COLOR        ds.b    1       * 22
PRW                ds.b    1       * 23
PATTERN_ADRS       ds.l    1       * 24
USER               ds.b    2       * 28

    .offset 0

```

リスト3 SAMPLE.S

```

    .include        iocscall.mac
    .include        doscall.mac
    .include        bems.h

    .xref   BEMS

    .xref   SDTBL
    .xref   DEFPTR
    .xref   COLCTL

    .text


    clr.l   -(sp)
    DOS     _SUPER
    move.l  d0,(sp)
    move.w  #256+14,d1
    IOCS    _CRTMOD
    moveq.l #2,d1
    IOCS    _B_CLR_ST
    IOCS    _B_CUROFF
    IOCS    _SP_INIT
    bsr     spdef
    bsr     sppal
    IOCS    _SP_ON

* DEFPTRテーブルの設定

    move.l  #$EBC000,DEFPTR         * B (使わないけど縁起物で)
    move.w  #0,DEFPTR+4
    move.l  #ENEMY,DEFPTR+8         * E
    move.w  #5,DEFPTR+12
    move.l  #MISSILE,DEFPTR+16      * M
    move.w  #1,DEFPTR+20
    move.l  #MY_SHIP,DEFPTR+24      * S
    move.w  #1,DEFPTR+28

* COLCTLテーブルの設定

    lea.l   COLCTL,a1
    move.b  #%00000000,(a1)         * B
    move.b  #%00000000,1(a1)        * E
    move.b  #%00000100,2(a1)        * MとEの衝突判定をする
    move.b  #%00000100,3(a1)        * SとEの衝突判定をする

* ベクトルテーブルの設定

    lea.l   ENEMY,a1
    move.w  #5-1,d0
    move.w  #32,d1
enemy_set_loop:
    move.b  #1,ENBL_CTRL(a1)                * E
    move.w  d1,POS_X(a1)
    move.w  #0,POS_Y(a1)
    move.b  #0,VECT_X(a1)
    move.b  #0,VECT_Y(a1)
    move.b  #18,SPEED(a1)
    move.b  #1,MOVE_DOT(a1)
    move.b  #1,CHRSIZE_X(a1)
    move.b  #2,CHRSIZE_Y(a1)
    move.w  #3,HITCHK_X1(a1)
    move.w  #4,HITCHK_Y1(a1)
    move.w  #12,HITCHK_X2(a1)
    move.w  #26,HITCHK_Y2(a1)
    lea.l   32(a1),a1
    add.w   #48,d1
    dbra    d0,enemy_set_loop

    lea.l   MISSILE,a1                      * M
    move.b  #1,ENBL_CTRL(a1)
    move.b  #0,VECT_X(a1)
    move.b  #-1,VECT_Y(a1)
    move.b  #40,SPEED(a1)
    move.b  #2,MOVE_DOT(a1)
    move.b  #1,CHRSIZE_X(a1)
    move.b  #1,CHRSIZE_Y(a1)
    move.w  #8,HITCHK_X1(a1)
    move.w  #10,HITCHK_Y1(a1)
    move.w  #9,HITCHK_X2(a1)
    move.w  #14,HITCHK_Y2(a1)

    lea.l   MY_SHIP,a1                              * S
    move.b  #1,ENBL_CTRL(a1)
    move.w  #128,POS_X(a1)
    move.w  #240,POS_Y(a1)
    move.b  #0,VECT_X(a1)
    move.b  #0,VECT_Y(a1)
    move.b  #25,SPEED(a1)
    move.b  #1,MOVE_DOT(a1)
    move.b  #1,CHRSIZE_X(a1)
    move.b  #1,CHRSIZE_Y(a1)
    move.w  #2,HITCHK_X1(a1)
    move.w  #10,HITCHK_Y1(a1)
    move.w  #13,HITCHK_X2(a1)
    move.w  #14,HITCHK_Y2(a1)
MAIN:
    bsr     BEMS
    tst.b   SDTBL
    bne     HIT                     * 衝突が発生した
    bsr     enemy_move
    bsr     ship_move
    bsr     gram_scroll
    bsr     check_ESC
    beq     MAIN
HIT:
    cmpi.b  #2,SDTBL+2              * M
    bne     SAMPLE_END
    movea.l SDTBL+8,a1
    clr.b   ENBL_CTRL(a1)
    clr.b   MISSILE+ENBL_CTRL
    eori.b  #1,flag
    beq     skip2
    move.w  #80,POS_Y(a1)
    cmpi.w  #128,POS_X(a1)
    bcs     skip1
    clr.w   POS_X(a1)
    bra     MAIN
skip1:
    move.w  #256,POS_X(a1)
    bra     MAIN
skip2:
    clr.w   POS_Y(a1)
    bra     MAIN
SAMPLE_END:
    DOS     _SUPER
    addq.l  #4,sp

    move.w  #-1,-(sp)
    move.w  #6,-(sp)
    DOS     _KFLUSH
    addq.l  #4,sp

    DOS     _EXIT

*++++++++++++++++++++++++++++++++++++++++
ship_move:
    lea.l   MY_SHIP,a1
    clr.b   VECT_X(a1)
    clr.b   VECT_Y(a1)
    move.w  #8,d1
    IOCS    _BITSNS
    btst.l  #7,d0                   * 4
    beq     up
    cmpi.w  #16,POS_X(a1)
    bcs     up
    move.b  #-1,VECT_X(a1)
up:
    btst.l  #4,d0                   * 8
    beq     right
    cmpi.w  #16,POS_Y(a1)
    bcs     right
    move.b  #-1,VECT_Y(a1)
right:
    move.w  #9,d1
    IOCS    _BITSNS
    btst.l  #1,d0                   * 6
    beq     down
```

▶先日，体育祭がありました。僕は１年生なので「ぐるぐる回ってえんやコーラ」という種目に出されてました。これはバットを軸にして15回まわり，30メートルほど走ってコーラを一気飲みする競技です。見ているほうはさぞ楽しかったことでしょう。

林　裕司(16)福島県

```
    cmpi.w  #256,POS_X(a1)
    bcc     down
    move.b  #1,VECT_X(a1)
down:
    btst.l  #4,d0
    beq     erase_missile                    * 2
    cmpi.w  #256,POS_Y(a1)
    bcc     erase_missile
    move.b  #1,VECT_Y(a1)
erase_missile:
    cmpi.w  #10,MISSILE+POS_Y
    bcc     check_missile
    clr.b   MISSILE+ENBL_CTRL                        * ﾐｻｲﾙを消す
check_missile:
    move.w  #6,d1
    IOCS    _BITSNS
    btst.l  #5,d0
    beq     ship_move_rts
    tst.b   MISSILE+ENBL_CTRL
    bne     ship_move_rts
    move.b  #1,MISSILE+ENBL_CTRL
    move.l  MY_SHIP+POS_X,d1
    move.l  d1,MISSILE+POS_X
ship_move_rts:
    rts
*+++++++++++++++++++++++++++++++++++++++++
enemy_move:
    lea.l   ENEMY,a1
    move.w  #5-1,d2
enemy_move_loop:
    move.b  #1,ENBL_CTRL(a1)
    move.l  MY_SHIP+POS_X,d0
    move.l  POS_X(a1),d1
    sub.l   d1,d0
    sub.l   #8,d0                            * y=y-8
    asr.w   #1,d0
    move.b  d0,VECT_Y(a1)
    swap    d0
    asr.w   #1,d0
    move.b  d0,VECT_X(a1)
    lea.l   32(a1),a1
    dbra    d2,enemy_move_loop
enemy_move_rts:
    rts

*+++++++++++++++++++++++++++++++++++++++++
check_ESC:
    moveq.l #0,d1
    IOCS    _BITSNS
    btst.l  #1,d0
    rts
*+++++++++++++++++++++++++++++++++++++++++
gram_scroll:
    lea.l   $e8001a,a1
    addq.w  #1,counter
    cmpi.w  #20,counter
    bcs     gram_scroll_rts
    clr.w   counter
    move.w  page_y,d0
    subq.w  #1,d0
    andi.w  #511,d0
    move.w  d0,page_y
    move.w  d0,(a1)
    move.w  d0,4(a1)
    move.w  d0,8(a1)
    move.w  d0,12(a1)
gram_scroll_rts:
    rts
*+++++++++++++++++++++++++++++++++++++++++
spdef:
    moveq.l #0,d1                            * パターンコード
    moveq.l #4-1,d7
    lea.l   pat0,a3
    lea.l   pat0+8,a4
spdef0:
    lea.l   buffer,a5
    movea.l a3,a2
    bsr     sp_def_sub
    movea.l a4,a2
    bsr     sp_def_sub
    moveq.l #1,d2
    lea.l   buffer,a1
    IOCS    _SP_DEFCG                        * パターン定義
    addq.l  #1,d1
    lea.l   256(a3),a3
    lea.l   256(a4),a4
    dbra    d7,spdef0
    rts

sp_def_sub:
    move.w  #16-1,d6
_spdef0:
    move.w  #4-1,d5
spdef1:
    move.b  (a2)+,d0
    lsl.b   #4,d0
    or.b    (a2)+,d0
    move.b  d0,(a5)+
    dbra    d5,spdef1
    addq.l  #8,a2
    dbra    d6,_spdef0
    rts
*+++++++++++++++++++++++++++++++++++++++++
sppal:
    lea.l   palet_data,a0
    lea.l   $e82220,a1
    move.w  #16-1,d0
sppal0:
    move.l  (a0)+,(a1)+
    dbra    d0,sppal0
    rts

    .data
counter:
    dc.w    0
page_y:
    dc.w    0
flag:
    dc.b    0
    .even

MY_SHIP:                                     * S
    ds.b    22
    dc.b    2                                * ﾊﾟﾚｯﾄﾌﾞﾛｯｸ2
    dc.b    3
    dc.l    SHIP_PAT
    ds.b    4
MISSILE:                                     * M
    ds.b    22
    dc.b    1                                * ﾊﾟﾚｯﾄﾌﾞﾛｯｸ1
    dc.b    3
    dc.l    MISSILE_PAT
    ds.b    4
ENEMY:                                           * E
    ds.b    22
    dc.b    1                                * ﾊﾟﾚｯﾄﾌﾞﾛｯｸ1
    dc.b    3
    dc.l    ENEMY_PAT1
    ds.b    4

    ds.b    22
    dc.b    1                                * ﾊﾟﾚｯﾄﾌﾞﾛｯｸ1
    dc.b    3
    dc.l    ENEMY_PAT2
    ds.b    4

    ds.b    22
    dc.b    1                                * ﾊﾟﾚｯﾄﾌﾞﾛｯｸ1
    dc.b    3
    dc.l    ENEMY_PAT3
    ds.b    4

    ds.b    22
    dc.b    1                                * ﾊﾟﾚｯﾄﾌﾞﾛｯｸ1
    dc.b    3
    dc.l    ENEMY_PAT4
    ds.b    4

    ds.b    22
    dc.b    1                                * ﾊﾟﾚｯﾄﾌﾞﾛｯｸ1
    dc.b    3
    dc.l    ENEMY_PAT5
    ds.b    4
*---------------------------------------------
*   dc.b    ｽﾌﾟﾗｲﾄ番号,ﾊﾟﾀﾝ番号

SHIP_PAT:
    dc.b    0,3

MISSILE_PAT:
    dc.b    1,2

ENEMY_PAT1:
    dc.b    2,0
    dc.b    3,1
ENEMY_PAT2:
    dc.b    4,0
    dc.b    5,1
ENEMY_PAT3:
    dc.b    6,0
    dc.b    7,1
ENEMY_PAT4:
    dc.b    8,0
    dc.b    9,1
ENEMY_PAT5:
    dc.b    10,0
    dc.b    11,1

*---------------------------------------------
pat0:
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,3,0,0,0,0,0,0,0,0,0,0,1,0,0
    dc.b    0,0,5,0,0,0,0,1,3,3,0,0,0,4,0,0
    dc.b    0,0,5,0,0,0,2,1,3,3,1,0,0,5,0,0
    dc.b    0,0,3,1,0,2,2,1,3,13,2,0,0,5,0,0
    dc.b    1,0,1,5,0,2,3,1,3,13,2,13,1,3,0,0
    dc.b    3,0,0,5,2,14,3,1,4,5,14,10,5,1,0,0
    dc.b    3,0,0,1,2,14,3,1,4,5,14,9,1,0,0,1
    dc.b    5,0,0,0,14,3,3,1,5,13,2,9,10,0,0,3
    dc.b    3,1,0,0,14,3,1,3,5,4,2,10,9,0,0,5
    dc.b    1,5,0,1,14,3,1,4,5,2,4,12,9,0,3,2
    dc.b    0,3,1,1,14,3,2,5,4,2,5,12,10,0,5,0
    dc.b    0,1,5,2,14,3,2,5,4,2,5,5,12,3,2,0
    dc.b    0,0,1,2,4,3,2,4,4,2,5,5,12,3,0,0
pat1:
    dc.b    0,0,0,2,4,3,2,4,2,3,5,5,12,0,0,0
    dc.b    0,1,0,2,4,4,2,4,2,4,5,12,11,0,1,0
    dc.b    2,4,0,2,3,4,3,2,3,5,5,11,13,0,5,2
    dc.b    5,4,0,4,14,14,3,2,3,1,11,9,4,0,3,5
    dc.b    2,3,5,14,2,1,2,2,2,2,1,2,0,3,5,0
    dc.b    0,4,0,0,0,14,14,5,5,12,12,0,0,0,5,0
    dc.b    0,4,0,0,2,14,5,5,10,8,10,12,0,0,5,0
    dc.b    0,4,0,0,14,14,5,5,8,6,8,12,0,0,5,0
    dc.b    0,2,1,0,14,4,4,5,10,6,10,5,0,1,2,0
    dc.b    0,0,4,0,0,14,4,4,5,5,5,0,0,5,0,0
    dc.b    0,0,3,1,0,0,2,3,3,2,0,0,1,3,0,0
    dc.b    0,0,0,4,1,1,4,0,0,4,1,1,5,0,0,0
    dc.b    0,0,0,1,3,4,1,0,0,1,5,3,1,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
pat2:
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,15,15,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,15,15,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,15,15,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,15,15,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,15,15,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,15,15,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,0,15,15,0,0,0,0,0,0,0
pat3:
    dc.b    0,0,0,0,0,0,0,14,14,0,0,0,0,0,0,0
    dc.b    0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0
    dc.b    0,0,0,9,11,12,2,1,1,2,12,11,9,0,0,0
    dc.b    0,12,13,11,11,11,2,10,10,2,11,11,11,13,14,0
    dc.b    0,14,13,13,13,11,12,10,10,12,11,13,13,14,14,0
    dc.b    0,0,11,12,13,13,13,13,13,13,13,13,14,7,0,0
    dc.b    0,0,5,5,11,12,14,14,14,14,7,7,7,7,0,0
    dc.b    0,0,5,6,6,6,7,7,7,7,7,7,7,7,0,0
    dc.b    0,0,5,5,6,7,6,6,7,7,6,7,7,7,0,0
    dc.b    0,0,5,6,5,7,7,6,6,6,7,7,7,7,0,0
    dc.b    0,0,5,5,6,5,6,7,6,6,7,6,7,7,0,0
    dc.b    0,10,5,5,5,6,6,6,7,6,6,7,7,7,10,0
    dc.b    0,14,13,5,6,5,6,6,6,6,6,6,7,13,14,0
    dc.b    0,13,13,5,5,6,5,6,6,7,6,7,7,13,13,0
    dc.b    0,0,11,13,13,14,14,14,14,14,14,13,13,11,0,0
    dc.b    0,0,0,0,10,12,12,13,13,12,12,10,0,0,0,0
palet_data:
    dc.b    $00,$00,$03,$45,$03,$C9,$04,$C9
    dc.b    $05,$41,$05,$C1,$EF,$FB,$D7,$F5
    dc.b    $A7,$E7,$87,$E1,$4F,$D3,$3F,$CF
    dc.b    $56,$95,$45,$D1,$04,$41,$84,$21
    dc.b    $00,$00,$07,$C0,$0D,$82,$00,$10
    dc.b    $6B,$5A,$B5,$C0,$DE,$80,$EF,$40
    dc.b    $10,$84,$31,$8C,$52,$94,$7B,$DE
    dc.b    $B5,$AC,$DE,$F6,$FF,$FF,$42,$11

buffer:
    ds.b    128

    .end

```

考え方と実例を探って

オブジェクト指向を取り入れる

Tan Akihiko 丹 明彦

性能至上で様式にとらわれないゲームプログラミングもオブジェクト指向という考え方を導入することで大きく様変わりします。ここではその基本的な考え方から見つめ直してみましょう。

オブジェクト指向ってなんだろう。なんでもそれを使えばプログラムの生産性がとても上がるらしい。まるで魔法のようだな。便利そうに思える。で，我々は結局なにをすればいいのか。ここでたいていの人は行き詰まる。優れた思想でも実現する手段を欠けば机上の空論である。人は忙しいのでついつい手近な環境の中でできることに閉じこもってしまう。

＊　＊　＊

もう5年も前のことになるが，本誌で「実用(?)オブジェクト指向のゲームプログラミング」という連載があった。ゲームプログラミングにはオブジェクト指向が向いているとし，オブジェクト指向でゲームプログラミングにアプローチする手法を示したものである。

当時は内容が高度すぎるという評判がなきにしもあらずだったが，いま読み返すとしごくまっとうな内容である。その冒頭を引用しよう。

「あの人騒がせなオブジェクト指向という言葉が至るところで使われるようになって久しいが，実際にそれがどういうものなのかといったイメージを持っている人は少ないだろう。なんだか得体のしれないものという印象が強いようだ」

残念ながら5年たったいまでもこの言葉は真実のようである。

さらに続けよう。

「ここはひとつ，具体的にゲームプログラミングというものに的を絞ってオブジェクト指向を適用することを考えてみたいと思う。実はコンピュータゲームというのは最もオブジェクト指向に向いたジャンルのひとつといわれているのだ」

そして著者の浜口氏はアドベンチャーゲームやシューティングゲームもオブジェクト指向でプログラムすれば計り知れないメリットを享受することができるといっているのである。

ソフトウェア危機

オブジェクト指向が台頭してきた背景として，いわゆるソフトウェア危機とそれを回避するための開発環境の必要性がある。ハードウェアの進歩は凄まじい。毎年倍々の勢いで速くなるCPUに，広くなるメモリ。当然，ゲームの規模は大きくなり処理は複雑化する。それが面白いゲームのために必要だと信じられているからだ。しかしプログラマの能力は倍々になったりはしない。

巨大化するゲームプログラムを作りおおせるためには，よほどパワーのあるプログラマでないと太刀打ちできない。ソフトウェア危機などといわれているのもそのへんのこと。そこにオブジェクト指向を柱とした開発環境を上手に導入することで，それほど能力のないプログラマでもそこそこのプログラムが組め，能力のあるプログラマならもっと素晴らしいプログラムを組むことができるようになるというのだ。

オブジェクト指向

オブジェクト指向の書物をあさると，必ずといっていいほどクラス，インスタンス，メソッド，メッセージ，インヘリタンス(継承）などの用語を理解することを要求される。どれもこれも抽象的な概念である。概念だけでぐいぐいと議論を押していける人ならそうやって理解することもできるが，世の中そういう人ばかりじゃない。

オブジェクト指向はデータの抽象化を推し進めたもの，といってピンとくる人などそうはいないに違いない。恥ずかしながら私だってピンとくるわけではない。

こういうことだと思っていただきたい。変数は，最初は独立した数値にすぎなかった。やがて変数は構造を持つようになり，その構造を代表する名前で構造全体をひとつの変数のように扱えるようになった。そしてその構造は自らの振る舞い，つまり処理をも自らの中に取り込み始めた。これがオブジェクト指向でいうところのオブジェクトである。

オブジェクト指向の思想をいくら唱えてもオブジェクト指向プログラミングができるようになるわけではない。実現するためには技術的なアプローチも必要だろう。

「実用(?)オブジェクト指向のゲームプログラミング」でもうひとつ特徴的だったのは，特殊なオブジェクト指向言語を使わずにZ80のアセンブラでオブジェクト指向を実現したところである。オブジェクト指向プログラミングはプログラミング言語でなく作法の問題なのである。どの言語を選ぶとよいとかどういう書き方をするのがオブジェクト指向らしいとかいう問題ではないのである，きっと。

プログラムを部品化する

オブジェクト指向のメリットとして「プログラムの部品化」がよくいわれる。

シューティングゲームを例にとろう。だいたいのシューティングゲームにおいて，基本的なルールは次のとおりになる。

- 自機はジョイスティックで移動する
- 敵機はアルゴリズムに従って移動する
- 自機は弾丸を発射する
- 敵機は弾丸を発射する
- 自機の弾丸は敵機を壊す
- 自機の弾丸は自機を壊さない
- 敵機の弾丸は自機を壊す
- 敵機の弾丸は敵機を壊さない

ほかにも敵機の出現するタイミングとか硬い敵とかボスキャラなどがあるが，それは割愛する。

＊　＊　＊

シューティングゲームを通常のプログラミング手法で実現すると，

(1-1)プレイヤーの操作を受け取る
(1-2)レバー操作なら自機を動かす
(1-3)ボタン操作なら弾丸を発射する

(2-1)敵機を動かす
(2-2)自機と敵機との当たり判定をする
もし当たっていれば自機は壊れる
(2-3)敵機から弾丸を発射する

(3-1)自機の弾丸を動かす
(3-2)敵機と自弾との当たり判定をする
もし当たっていれば敵機は壊れる

(4-1)敵機の弾丸を動かす
(4-2)自機と敵弾との当たり判定をする
もし当たっていれば自機は壊れる

これを順番に繰り返す。まっとうな方法である。このやり方でシューティングゲームは作れるし，事実作ってきた。ではなにが悪いのか。

敵機の種類をひとつ増やすことを考えると，このやり方の辛さがわかる。(2-1)から(2-3)を増やすのは当然として，(3-2)も増やさなくてはいけない。

2人同時プレイをする場合を考えるともっとよくわかる。「自機との当たり判定」と書いてあるところを「自機その1との当たり判定」と「自機その2との当たり判定」に全部書き直さなくてはならない。それをすべての種類の敵機についてやることを考えると気が狂いそうになる。

確かに強力で処理の効率もよいのだが，プログラムする人間に大きな負担を強いるのである。ゲームの規模が巨大になるほどこの傾向は顕著になる。これがゲームのプログラミングにおけるソフトウェア危機の端的な例である。

＊　　＊　　＊

では同じシューティングゲームをオブジェクト指向でプログラムするとどうなるか見てみよう。

オブジェクト指向においては，登場するものすべてがオブジェクトである。シューティングゲームの場合は自機，敵機，自機の弾丸，敵機の弾丸などをオブジェクトとして表現する。以下にそれらの動作を定義してみよう。

●自機

(5-1)ジョイスティックから入力を得る
(5-2)レバー入力なら移動する
(5-3)ボタン入力なら弾を発射する
(5-4)他オブジェクトとの衝突判定をする
(5-5)もし衝突していたら相手オブジェクトに対して衝突メッセージを送る
(5-6)敵機または敵弾から衝突メッセージがきたら壊れる

●敵機

(6-1)アルゴリズムに従って移動する
(6-2)アルゴリズムに従って弾を発射する
(6-3)他オブジェクトとの衝突判定をする
(6-4)もし衝突していたら相手オブジェクトに対して衝突メッセージを送る
(6-5)自弾から衝突メッセージがきたら壊れる

●自弾

(7-1)アルゴリズムに従って移動する
(7-2)他オブジェクトとの衝突判定をする
(7-3)もし衝突していたら相手オブジェクトに対して衝突メッセージを送る

●敵弾

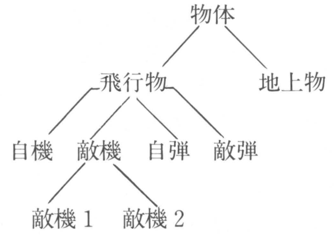
(8-1)アルゴリズムに従って移動する
(8-2)他オブジェクトとの衝突判定をする
(8-3)もし衝突していたら相手オブジェクトに対して衝突メッセージを送る

以上のような動作を定義したオブジェクトを適切なタイミングで呼び出していく。メッセージは特徴的な概念である。衝突した相手に対して発するのだが，もしここで敵か味方かという程度の情報しかメッセージに込めなければ，破壊されたかどうかの判定は極めて単純に行える。敵機の種類を増やしても，必要な処理は変わらない。ほかのオブジェクトに影響が及ばない。これはかなりのメリットである。

＊　　＊　　＊

上の4種類のオブジェクトの動作には共通点がある。ある規則によって移動し，衝突した相手にメッセージを送るという点である。動作の記述もそっくりで，一部だけ異なるにすぎない。

登場キャラクタの関係をオブジェクト指向風に書くと次のようになる。



自機も敵機も自弾も敵弾も，「飛行物」と分類できる。飛行物というオブジェクトの実体が登場するわけではない。飛行物は「ある規則によって移動し，衝突した相手にメッセージを送る」という動作を定義された物体，という概念である。このような，オブジェクトの動作を規定するものをオブジェクト指向ではクラスと呼んでいる。

そして自機や敵機などは，飛行物クラスからその性質を継承しつつ，それぞれに特有な性質を追加したサブクラスとして規定される。

敵機は1種類ではないので，敵機も一般的なクラスとして定義し，移動のアルゴリズムや外見などの異なる複数の敵機サブクラスを作る。

こうして登場キャラクタの動作を規定するクラスがそろったところで，それらの実体，インスタンスを作る。ここで初めてキャラクタはゲームの中に登場する。

また，ゲームには飛行物のほかに地上物などが登場する。そこで飛行物クラスと地上物クラスのさらに上位にあるクラス，「物体」クラスを定義する。物体クラスはどのクラスにも共通なもの，ここでは座標を持っている。

＊　　＊　　＊

クラス階層がはっきりしたところで，クラスを定義してみよう。インスタンスを生成するクラスとそうでないクラスがある。

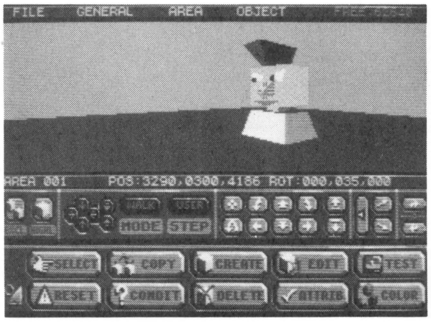
物体：
座標を保持する。
飛行物：
物体クラスのサブクラスである。
衝突したオブジェクトに対してメッセージを送る。
地上物：
物体クラスのサブクラスである。
地上に拘束されている。
衝突したオブジェクトに対してメッセージを送る。
自機：(インスタンスを生成する)
飛行物クラスのサブクラスである。
ジョイスティックから入力を受け取る。
敵機または敵弾で破壊される。
自弾を発射する。
敵機：
飛行物クラスのサブクラスである。
自弾で破壊される。
敵弾を発射する。
敵機1：(インスタンスを生成する)
敵機クラスのサブクラスである。
直線状に移動する。
敵機2：(インスタンスを生成する)
敵機クラスのサブクラスである。
ジグザグに移動する。
自弾：(インスタンスを生成する)
飛行物クラスのサブクラスである。

敵弾：(インスタンスを生成する)
　飛行物クラスのサブクラスである。

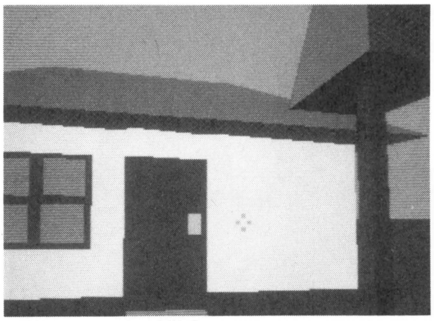
　ここで「プログラムの部品化」の話に戻る。クラスと継承は，上手に使えば，プログラムを部品として蓄積することを可能にする。

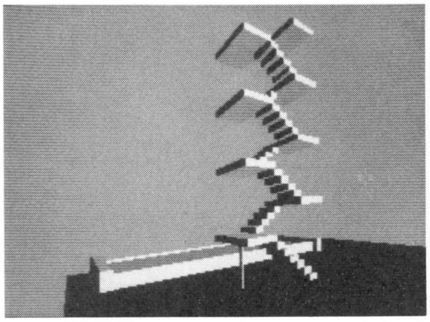
　たとえば，敵機の種類を増やすには，敵機クラスから継承した新しい「敵機3」クラスを作り，敵機3特有のアルゴリズムを盛り込めばよい。2人同時プレイにしても，自機クラスが用意されているわけだから，インスタンスをひとつ余計に作るだけ。新しい種類のキャラクタがほんのちょっとの手間で定義できるのである。

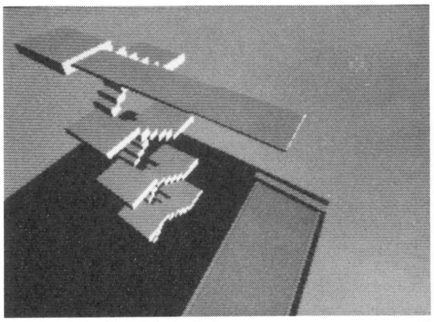
　オブジェクトは，そのオブジェクト自身に関するすべての情報(データやアルゴリズムなど)を自分自身で管理しているので，ひとつオブジェクトを増やしても周囲に与える影響が最小限ですむ。



エディタの基本画面


WALKモードはVR感覚？


まさに積木感覚のエディタだ

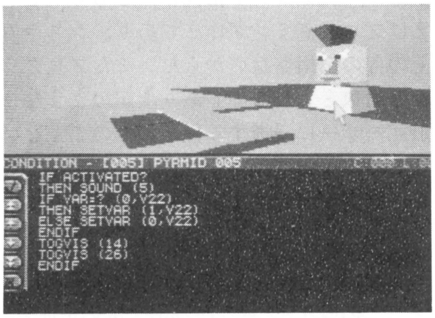

FLYモードだと視点も自在

ゲームプログラミング環境の近未来
——VIRTUAL REALITY STUDIO——

　シューティングゲームのキャラクタはそれぞれが独立に意志を持って動いているように見える。上の例で示したとおり，その処理の正体はそれぞれのキャラクタを順番に少しずつ動かすことにすぎない。プレイヤーの目には全部のキャラクタが同時に動いているように見える。キャラクタたちは本質的に並列動作するものであり，クラス単位で動作を定義できるオブジェクト指向との相性がよい。

　というところで今回ぜひとも紹介したいソフトウェアの話に強引に持っていく。本誌1992年9月号のTREND ANALYSISで簡単に紹介した「VIRTUAL REALITY STUDIO」である。例によって海外作品で,AMIGAとIBM PCにしかないのが残念。

　VIRTUAL REALITY STUDIOは，もともとヨーロッパで「3D CONSTRUCTION KIT」という名前で発売されたアドベンチャーゲーム制作ソフトウェアである。拡大解釈すれば，バーチャルリアリティに見えないこともないのでこういう名前になったのだろう。さらに私は拡大解釈し，このVIRTUAL REALITY STUDIOに，オブジェクト指向と並列処理の要素を見た。



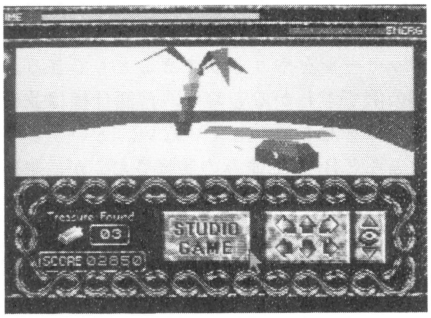
物体ごとにプログラム可能

　冷静に見れば，1つひとつの機能は低い。しかしVIRTUAL REALITY STUDIO環境下のプログラミングスタイルはいままで経験のなかったものである。

＊　　＊　　＊

　VIRTUAL REALITY STUDIOを立ち上げると，ユーザーは計算機の中の仮想空間に出現する。ユーザーは仮想空間の中に立方体や三角形の板などの物体を作ることができる（このエディタ部，大きさなどを変更すると画面中でポリゴンの物体がリアルタイムでぐいぐい動く優れものだ）。

　この物体は仮想空間内を自由に動かすことができるが，物体同士が衝突するとそれ以上は動かない。干渉チェックによって物体同士がめり込むことを防いでいる。



サンプルでついてくるゲーム

　こうして物体を積み上げて作った仮想空間の中を歩き回ることができる(ウォークスルー)。壁にぶつかると通り抜けることはできないし，階段を作っておけば上っていく。入出力をそれらしくすればいっぱしのバーチャルリアリティである。

　しかし今回強調したいのは，このシステムのオブジェクト指向的な側面である。仮想空間の中に作った物体の1つひとつに対してプログラムできるのだ！　例を示そう。

例1：撃つと消える立方体

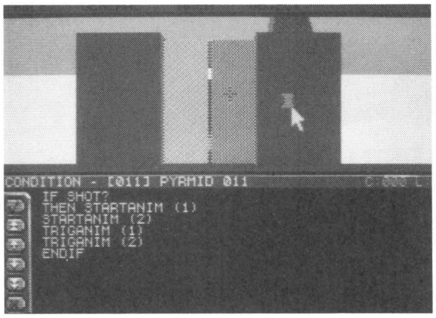
　ユーザーはVIRTUAL REALITY STUDIOの仮想空間内を歩き回りながら，物体に照準を合わせて撃つことができる。「撃った」という情報は狙いをつけたオブジェクトに伝えられる。この時点で物体に登録されているプログラムを起動する。この物体に「撃たれたら……」という条件判定と処理をプログラムしておけば，撃つとなんらかの反応を示すオブジェクトができあがるというわけ。オブジェクト指向風にいえばメッセージによってメソッドを起動する，ということになる。

・立方体を作る。物体番号は2となった。
・立方体のプログラムエディタを開いて次のように書く。

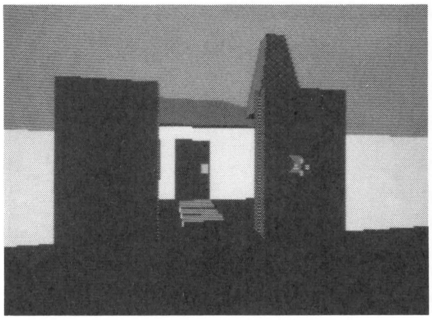
```
IF SHOT? THEN
   INVIS(2)
ENDIF
```

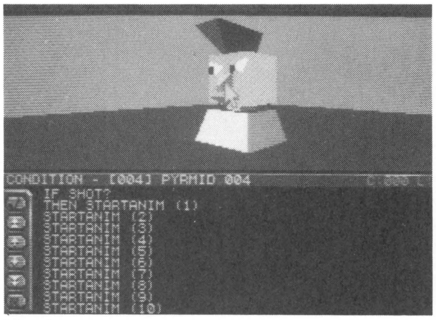
・プログラムエディタを閉じる。
・立方体を撃つと消える。

　プログラムした結果は即時に目の前の物

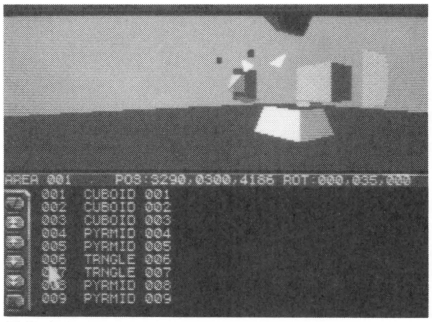


ボタンを押すと……


扉が開く



撃つと……



爆発する

体に反映するのである。

例２： 撃つとドアが開くボタン

・ボタンとドアになる直方体を作る。物体番号はそれぞれ3と4。

・ボタンのプログラムエディタを開いて次のように書く。

```
IF SHOT? THEN
   STARTANIM(1)
ENDIF
```

・プログラムエディタを閉じる。

・アニメーションのプログラムエディタを開いてアニメーションを記述する。アニメーション番号は1。

```
INCLUDE(4)
START
LOOP(10)
   MOVE(−10, 0, 0)
AGAIN
```

・プログラムエディタを閉じる。

・ボタンを撃つとドアがするすると開く。

＊　＊　＊

プログラムといえばテキストエディタで書くものと相場が決まっている。テキストエディタで書いたプログラムはテキストファイル，つまり１次元的につながる文字列のかたちで保存される。「プログラムリスト」っていうくらいだ。

VIRTUAL REALITY STUDIOのプログラムは少々異なっていて，物体にくっついている。画面から物体を選択してプログラムエディタを開き，その物体に関する処理を記述する。物体はもちろん1次元的に並んでなんかいないから，プログラムもリストのかたちでは表せない。仮想空間に浮かぶプログラムリストの断片というイメージ。

物体は自らのふるまいを自ら管理している。これはいうならばメソッドである。

また，ほかの物体を見えなくしたり，ほかの物体のプログラムを起動することもできる。これはメッセージといえる。

この意味で，VIRTUAL REALITY STUDIOの物体は単なる図形でない「オブジェクト」なのである。確かに，クラスも継承も備えてはいないし，正しくオブジェクト指向してはいないのだが，私はこれをあえてオブジェクト指向の環境と呼びたい。

＊　＊　＊

VIRTUAL REALITY STUDIOはリアルタイムで動作する環境である。ユーザーが仮想空間を歩き回っている最中も物体が飛び回る。もちろん複数の物体が好き勝手にアニメーションすることもある。

複数の物体が並列に動くというのはリアルタイムのシステムでは当たり前のことだが，VIRTUAL REALITY STUDIOの場合，ユーザーがプログラムしたオブジェクトも並列に動いている。オブジェクト間でメッセージをやりとりすることもできる。上の例でもわかるとおり，言語仕様は並列性をほとんど意識していない。というよりほとんどBASIC並みの手軽さだ。が，並列プログラミングの雰囲気は味わえる。

＊　＊　＊

最初にこのソフトで遊んだときはオブジェクトを積木のように並べていくだけのものかと思っていたが，さりげなくプログラミング作法の最先端を突っ走っていたりするのである。

とにかくプロトタイピングが楽。とりあえず形だけ作って仮想空間を歩き回る。そして物体にプログラムをくっつける。その物体を撃って反応を楽しむ。ひとつできたら次の形を作る。付け足し付け足ししていくうちに，いつのまにか巨大なゲーム空間ができているという感じ。「3Dデゼニランド」くらいは根性さえあればすぐだ。

プログラム言語そのものはそれほど進歩的ではない。オブジェクトやアニメーションを通し番号で指定するというのもイマイチ。そこは割り切っているのだ。これもまたチープでいい，ってもう欠点が欠点に見えていない。高尚な理論じゃない。現物がここにある。私にはそのこと自体が衝撃であった。

VIRTUAL REALITY STUDIOは，ひょっとすると，並列動作，イベント，オブジェクト指向などを備えた1990年代のBASICかもしれない。私たちがBASICを通じてプログラミングを学んだような，いい意味での入門用システムとしてのBASICである。程度は決して高いとはいえないが志は高い。

オブジェクト指向というキーワード

この世の中，人にものを説明するときにはキーワードが欠かせない。が，それがただのキャッチフレーズになってしまうとまた問題である。キャッチフレーズが独り歩きを始めると，そのおおもとの定義はどこかへ飛んでいき，拡大解釈がまかり通る。人工知能はパターンマッチング，ロールプレイングゲームは経験値稼ぎ，マルチメディアはCD-ROM，シミュレーションゲームはヘックス，バーチャルリアリティはデータグローブという具合だ。

さて，今回の主題である「オブジェクト指向」だが，上のような誤解をされているというわけではなさそうである。というよりも，オブジェクト指向は，どちらかといえばプログラミング作法であり，システムそのものを指さないからだ。またプログラマを対象としたものであり上記のような一般ユーザーに浸透する種類のものでもない。そして，十分に理解されたとはいいがたいし，誰もが認める定義があるわけでもない。

参考文献
実用(?)オブジェクト指向のゲームプログラミング――第1回 オブジェクトの正体を探る，浜口 勇，Oh!X1987年12月号pp.67-72

シナリオ管理の方法

アドベンチャーシステムを考える

Kousaka Tadashi 香坂 正嗣

単純な構造を持つプログラムと複雑なシナリオ，アドベンチャーゲームは比較的作りやすいプログラムです。ここではアドベンチャーゲームの要となるシナリオインタプリタを中心に考えてみましょう。

皆さん，ちょっとの間だけ目を閉じて，頭の中をからっぽにしてみましょう。なにか見えてきませんか？　それは，あなたの心の中にあるもうひとつの世界，不思議の国。今日はちょっと，この不思議の国を覗いてみましょう。

おや，ひとりの女の子が鏡に向かって話しかけていますね。

「鏡よ鏡，この世でいちばん難しくて，いちばん簡単なものはなあに？」

鏡は答えます。

「それは，アドベンチャーゲームのプログラミングです」

「あら，それはどうして？」

鏡は答える代わりに目のくらむような光を放ち始めました。とても目を開けていられません。……気がつくと，ここは不思議の国の入り口です。おやおや？　入り口に門番が立っていますねー。しかも門の脇に立て札が立っていて，こんなことが書いてあります。

「中に入りたい人は，門番となぞなぞ遊びをすること」

ちょうどいい。さっき鏡が答えてくれなかった，なぞなぞを出してみましょう。

「門番さん，この世でいちばん難しくて，いちばん簡単なものってなあに？」

あれれ，門番は考え込んでしまいました。ちょっと難しかったかな？

しかし，門番さんは答えます。

「いちばん難しいものだったらわかるけど……」

この際，ここを通れればなんでもいい！　それでいいから答えてもらいましょう。

「それは，アドベンチャーゲームのプログラミングだよ」

おっと，鏡が答えたのと同じ答え。こうなると，その理由が知りたくなります。

「簡単なことさ。普通のアクションゲームと違って，アドベンチャーゲームはシナリオが命だからだよ」

「？？？　よくわからない……」

「アドベンチャーゲームでは，ゲームのプログラミングのほかにシナリオの制作が必要になってくるじゃないか」

なるほど，ほかのゲームと違って，アドベンチャーゲームを作るには，プログラミングという作業のほかに，小説のように綿密なシナリオ作成をしなければゲームを完成させることはできません。なるほどこれは確かに難しい，うん。

しかし，門番さんはなぞなぞが解けたので，気分よく私たちを門の中へ入れてくれました。

まずはめでたし。

アドベンチャーゲームは難しい？

アドベンチャーゲームの命はシナリオであるといわれています。

先ほどの物語でもわかるように，アドベンチャーゲームの作成には綿密なシナリオ作成という難しい問題があります。これはプログラミング技術だけでは解決しない問題です。

シナリオにはゲーム中に起こることすべてが記述されていなければなりません。複雑なアドベンチャーゲームでは，必然的に多くの要素がからみあっていき，さまざまな条件の下での展開も記述しなければなりません。さらにシューティングゲームやパズルゲームよりも綿密なバックグラウンドを構築しなければなりませんし，文章の才能が要求されることもあるでしょう。

しかし，アドベンチャーゲームにとってシナリオはゲームプログラム本体以上に不可欠な存在なのです。

＊　　＊　　＊

でもちょっと待ってください。先ほどの話では，確か鏡さんは，この世でいちばん簡単なものもアドベンチャーゲームのプログラミングだっていってたはず……。いったいなにが簡単なのでしょうか？

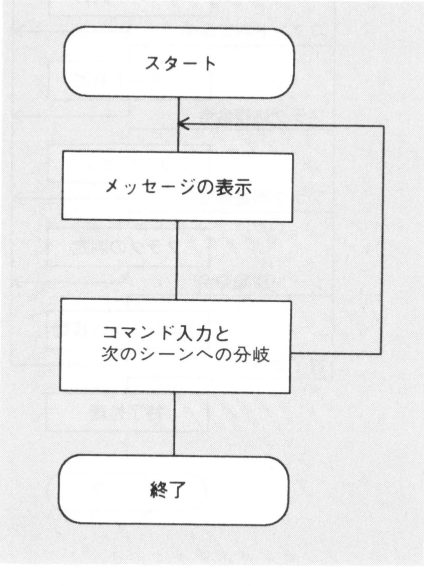
実は，突き詰めていくと，アドベンチャーゲームに必要な大きな要素はたった2つしかありません。ひとつはメッセージを表示すること。そしてもうひとつは，コマンドを入力して次のシーンへ進むこと。この2つの動作を繰り返していけば，アドベンチャーゲームは完成してしまいます（図1参照）。

もちろん，メッセージ表示にグラフィックが加わったり音楽やアニメーションなどの処理が付属することもあります。入力も，古典的タイプのアドベンチャーゲームではキーボードからの単語入力，最近はメニューからの項目選択その他という具合にユーザーインタフェイスのバリエーションはいろいろありますが，ゲームシステムにしてみれば，どんな方法によるものであれ入力は入力です。

結局，図1のループを作れば，アドベンチャーゲームができるわけです。なるほど。これなら簡単ですね。

でも，ちょっと待ってください。これに

図1



は，ひとつ重要な要素が抜け落ちています。これだけでは，まだ完全なアドベンチャーゲームではありません。いまの説明では，簡単なゲームブック程度のものしか作れないのです。もうちょっと改良してみましょう。

図２を見てください。

基本的なアドベンチャーゲームのプログラムは，

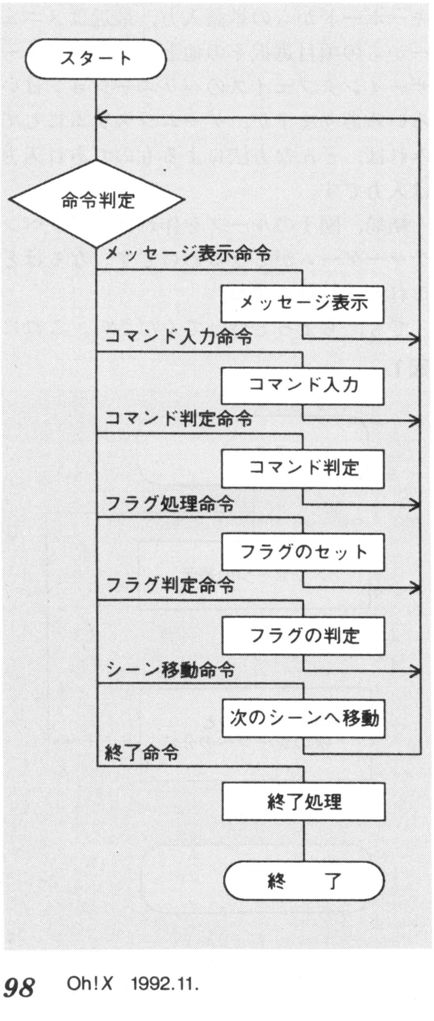
　メッセージ（または絵）を表示する
　コマンドの入力をする
　コマンドの判定をする
　フラグのセットをする
　フラグの判定をする
　次のシーンへ移動する

という要素の繰り返しで実行されています。フラグとは，プログラミングでお馴染みの「旗」のことです。YES/NOの情報を格納するためのもので，大きな意味では普通の変数もフラグとして扱われます。

あるシーンでナイフを拾ったら，「ナイフを持っているフラグ」がセットされるとか，一定のイベントをクリアしたら……などのように使用されます。

なんらかのアドベンチャーゲームをやっ

図2



たことのある人ならば容易に理解できるでしょう。特定のフラグがセットされているかどうかで処理を振り分けるのは，アドベンチャーゲームの基本になっています。

こういったフラグ関係の機能や，ひとつのシーンのなかでコマンドの入力や判定が何回もできるところがゲームブックとは違うところといえるでしょう。

しかし，こんなに複雑になってしまうと，ちょっと「アドベンチャーゲームは簡単」とはいえないような気もしますね。

シナリオインタプリタってなに?

BASICを知っている人なら想像がつくでしょう。結局のところ，アドベンチャーゲームの基本構造は，IF文の塊です。すなわち，

　一定の操作でフラグを変化させ
　フラグの状態によって状況を変える

ということに集約されます。

これをそのままIF文で組むと……あまり大規模なゲームはできないのはおわかりでしょう。

ここで登場するのが「シナリオインタプリタ」と呼ばれるものです。

ここでOh!MZの1987年７月号を開くと，怪しい記事が……。なになに，STORY MASTERだって？　ふーんアドベンチャーゲーム記述用言語かぁ。そう。シナリオインタプリタっていうのは，シナリオを記述するための簡易言語だったんですね。これさえあれば，プログラムを作るのと同じような感覚で，シナリオを書くことができるし，シナリオが完成したら，そのままゲームが完成するのですから，プログラミングの手間が半減するわけです。

STORY MASTERはその文法を見る限り，かなりBASICを意識した構成になっており，おまけにエディタを内蔵していたのでさらにBASICっぽい開発環境までも備えているものでした。しかし，通常，シナリオインタプリタといった場合，できあがったシナリオを解釈実行する部分だけを指すことが多いようです。

そう，基本文法体系に，みんなが知っている言語に近い記述方法を選ぶということは，新しく覚える知識が少なくてすむから，それだけ使い勝手がよくなります。これはうまい選択といえるでしょう。当時はBASICの普及率がほかの言語を圧倒していたから，BASICを基準にシナリオインタプリタを作成したのでしょうか。

さしずめ，現在ならＣ言語を参考にして作るのもいいかもしれません。特にＣ言語の関数的な考え方は，そのままアドベンチャーのシーン分けに応用できます。開発方法も，

　エディタでソーステキストを書く
　中間言語にコンパイルする
　シナリオインタプリタで実行する

という馴染みのスタイルになるのでしょうか。これから作成するならば，Ｃ言語をベースにしたシナリオインタプリタというのも面白そうです。

＊　　＊　　＊

では，問題です。

シナリオインタプリタって，どうやって

作ったらいいのでしょうか？

シナリオインタプリタの作り方

シナリオインタプリタの作り方のコツは，まず，必要最低限の機能をまとめて文法を決めることです。文法を決めたら，次にその文法にのっとって自分で簡単なプログラムを実際に書いてみることです。この過程で，仕様の不備な点や必要ない機能などを洗い出すことができます。

サンプルとして作成してみたのが，リスト１のファイルです。Ｃ言語を手本にした簡単なシナリオ記述言語の例です。Ｃのプログラムが読める人なら，苦労なしに読めると思います。

最初にフラグ変数の定義をしていますね。フラグ変数とは，０か１しか入らない変数のことです。アドベンチャーゲームでは，それ以上の大きさの数を扱うことはきわめて稀ですので，たいていこれで間に合うでしょう。どうしてもそれ以上の数を扱いたいときは，変数の種類を増やせばいいこと。Ｃの変数もchar型やint型といった，複数の変数型を持っているのですから，それを参考にすればいいと思います。このように変数の型を加えれば，アドベンチャー用言語は簡単にロールプレイング用言語にもなります。

さて，ここではkeyというフラグを定義しています。この変数にはゲーム中に鍵を手に入れたら１，そうでなければ０，が入るようになっています。

その下のinventoryという命令は，フラグ変数keyを持ちものを扱うフラグとして定義しています。持ちもののチェックを行ったとき，このフラグが１なら“かぎ”という文字を表示するためのものだと思ってください。

アドベンチャーに持ちもののチェック機能と，データのセーブ/ロードの機能は不可欠です。これらの機能はインタプリタ本体で用意しておくのがいいでしょう。

msgはメッセージ表示の命令，menuはコマンドメニューの表示と入力の命令です。コマンドを入力したあとはswitch，case文でメニューの入力別の処理を行います。

プログラム中にときどき見える(key)や(!key)という命令は，フラグの条件判断命令で(key)は，key＝１のときに以降の文を実行し，(!key)はkey＝０のときに以降の文の実行を行う命令です。

さて，以上が文法の簡単な説明です。一応，叩き台として，こんな文法でこれくら

表１　STORY MASTERの文法

変数	数値変数	#A0，#A1……#B0……#Z9　260個 #で始まる３文字の文字列は数値変数を示す。変数はＡ～Ｚの英大文字で26種類に区分され，さらにそれぞれ０～９の数字をつけて全体で260個に分類される。これらの変数はその中に０～99の符号なし変数を代入することができる。また，変数名の数字の省略を行った場合，数字は０を指定したものとする。
変数	文字変数	＄０，＄１……＄９　10個 ＄で始まる２文字の文字列は，文字変数を示す。文字変数は０～９の数字を直後に添えることによって，10種類に分けることができる。これらの文字変数は，16文字以下の短い文字列を格納することができる。また，変数名の数字を省略すると数字は０を指定したものとする。
変数	コマンドワーク	COM１，COM２　２個 COM１とCOM２は，INPUT文により入力されたコマンドを２つの単語に分割して，セットする一種の文字変数である。ただし，コマンドワークはそれ自体で直接，表示や比較を行うことができないため，一度文字変数に代入しなくてはならない。
ステートメント	BEEP　n BEEP　１	ベルを鳴らす。ｎは回数。
ステートメント	END	プログラムの終了。
ステートメント	GOTO ［LABEL］ GOTO ［SCENE　１］	指定したラベルへジャンプ。
ステートメント	GOSUB ［LABEL］ GOSUB ［INVENTORY］	指定したラベルをサブルーチンコールする。
ステートメント	IF式～ IF#A >？“ABC” IF＄＝“ABC”GOTO ［SCENE　１］	式は比較を行う。比較に使うことのできる記号は＝，<，>，<>の４種類。ただし文字変数の比較のときは＝しか使えない。
ステートメント	INPUT	キーボードからコマンドを入力する。コマンドは２分割されて，COM１，COM２にセットされる。
ステートメント	PASS PASS “ADVENTURE２”	新しいプログラムをロードして実行する。
ステートメント	PAUSE/！	“HIT ANY KEY”と表示してキー入力待ちになる。「！」は省略形である。
ステートメント	PRINT/？～式 PRINT　C　……… PRINT　H　……… PRINT“ABC”　……… PRINT　#A　……… PRINT　＄１　……… PRINT[MSG1]　……… PRINT“#A＝”；#A　………	式に従って画面に出力する。「？」は省略形 画面クリア カーソルホーム 文字列表示 数値変数表示 文字変数表示 ラベルからの文字列表示 式と式をつなげる
ステートメント	REM/' REM＊＊＊＊＊INIT.＊_＊＊＊	注釈文定義。「'」は省略形
ステートメント	RET	GOSUBによるサブルーチンコールを終了して，メインプログラムに復帰する。
ステートメント	USR（adr変数） USR（IFC１　#A USR（IFE５　＄９）	マシン語で作ったプログラムに実行を移す命令。その際パラメータに変数を使うことが可能で，変数が数値変数の場合は，Ａレジスタに変数の内容，DEレジスタに変数の格納番地を，文字変数の場合は，DEレジスタに変数の格納番地をセットして，指定した番地をコールする。マシン語プログラム中のRET命令でインタプリタに復帰する。
ステートメント	／ PRINT “ABC”／	改行を行う。
ステートメント	（COM１ COM２） （LOOK DOOR)？“OK”#A＝１ （EAT＊）GOTO ［EAT］	INPUT文で，COM１，COM２にセットされたデータとの比較を行う。COM２に単語の代わりに「＊」を定義したとき，COM１が等しければ，それ以降の１行を実行する。COM１に「＊」を使ったときも同様。
ステートメント	<文字列> <INPUT (LOOK＊)GOSUB[LOOK]> PRINT “??”>	固定ループ。プログラム中で「>」を発見すると，その直後に実行した「<」のところまで戻る。ただし，この命令自身は無条件でループを行うためループ内でプログラムの分岐を行わなくてはならない。またそのためループを重ねてもネスティングは深くならず，最初に定義したループは，無視される。
関数	INC変数 INC#A	変数の値に１加算する。
関数	DEC変数 DEC#A	変数の値から１引く。
関数	RND（式）変数 RND（10）#A	０～式－１の範囲の乱数を変数にセットする。
データ	［文字列］ ［SCENE　１］	ラベル 場面の定義やメッセージの表示などで指定される。［　］で囲み，行頭に置くことにより定義される。
データ	：単語１　単語２　単語３… ：LOOK SEE MIRU ：TAKE GET TORU HIROU	辞書データ 「：」（コロン）の後ろに続けて単語の同意語を指定する。
データ	“文字列” ［MSG］“[[INPUT COMMAND]]”	メッセージデータ ラベルと一緒に指定する。PRINT文で，メッセージのラベル指定を行うとき，この方法でデータをプログラム中にセットする。
エラー	Syntax　Error	文法エラー。
エラー	No Label　Error	指定したラベルがない。
エラー	Nesting　Error	GOSUB文のネスティングが４重を超えた。
エラー	Bad loop　Error	<……>のループで該当する「<」がない。
エラー	RET without GOSUB	GOSUB文に対応しないRET文を実行した。

いの命令があれば，シナリオインタプリタとして機能すると思います。あと，絵を表示したければ，PICファイルやCUTファイルのロード命令でも付け足せばいいでしょう。PICLIBやAPICLIBなどのライブラリを使えば簡単に作ることができます。

さて，あとは気力と体力の勝負。スクロールシューティングゲームを作るのと違い，CRTのタイミングを気にする必要もなければ，速いアルゴリズムもいりません。シナリオインタプリタのよいところは，最低限の技術さえあれば作ることができるところなのです。

シナリオについて

アドベンチャーゲームは，ゲームの進行方法によって，いくつかの種類に分類できます。図3にその例がいくつかあります。

まずは，もっともオーソドックスな紙芝居型。これは，シーン（場面）が鎖状にひとつずつつながっていて，ゴールまで一直線にシナリオが展開されます。この形式のいいところは，作り手が理解しやすいこと。悪いところは，どんな人がプレイしても解決するルートはひとつしかない点です。この形式はもっとも古典的な形式で，アドベンチャーゲームの基本中の基本となります。まずは，このかたちを基本として入るのが簡単でよいでしょう。

２番目は神経衰弱型です。トランプの神経衰弱と同じように，プレイヤーがすべてのシーンにいつでも自由に出入りできるものです。最近はこの手のシナリオ展開をするゲームが増えてきました。この形式のいいところは，自由に移動ができるため，より，リアルなシナリオが作れること。悪いところは，作り手も，プレイヤーもシナリオを理解しづらいことです。

３番目がその２つの複合型。ある程度のシーンをひとつにまとめたシーンの集合が，１列に並んだもので，プレイヤーはいろいろなシーンを行き来しながら，条件を満たして次のステップへと進んでいくというものです。紙芝居型と神経衰弱型のいいところをうまくくっつけた形式ですが，この複合型がいちばん理想的なアドベンチャーのシナリオ形式だと思います。

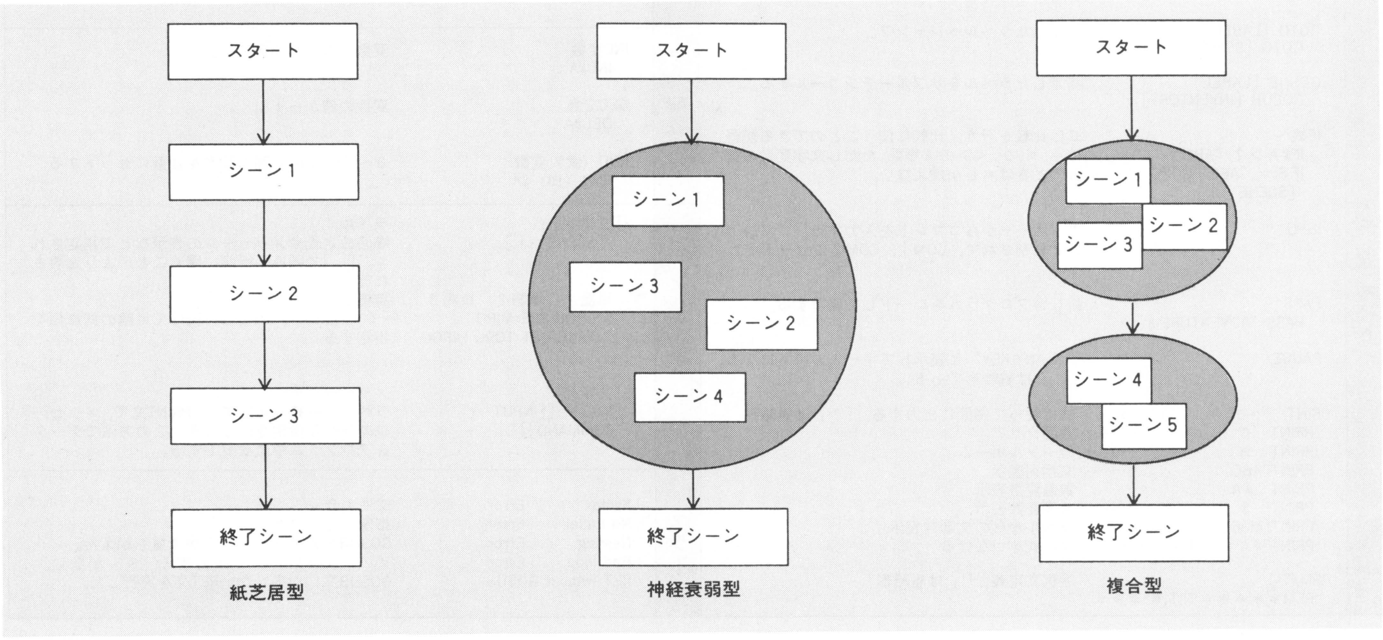
＊　　＊　　＊

市販のゲームを見ていると，最近のアドベンチャーゲーム事情は，あまりいいものとはいえません。でも，まるで小説を読む感覚で遊べる手軽さは，ほかのゲームではちょっと味わうことができないと思います。もっとコクがあって，それでいて上品な味わいのアドベンチャーゲームが現れる日を待ちながら，不思議議の国をあとにすることにいたしましょう。

リスト1

```
flag        key;
inventory  key="かぎ";

scene   room
{

    msg( "部屋の中にいる" );       * メッセージの表示

    menu(#1)={                     * メニューの表示、選択
             1:"見る",
       (!key)2:"取る",
             3:"移動する"
    }

    switch( #1 )
    {
        case  1:                   *メニューで「見る」を選択
                   msg( "大きな部屋の中です。テーブルとドアが見えます" );
              (!key)msg( "テーブルの上に鍵があります" );

        case  2:                   *メニューで「取る」を選択
             msg( "OK!  鍵を手に入れました" );
             key = 1;

        case  3:                   *メニューで「移動する」を選択
              (!key)msg( "鍵が無いので、ドアが開きません。移動できない" );
               (key){
                   msg( "OK!ドアを開けて部屋の外に出ます" );
                   scene = outdoor;
               }
    }
}

scene   outdoor
{
    msg( "あなたは部屋の外へ出ました" );
    end;
}
```

図3



プログラム速度管理関数

XVI.FNC

Kamiyama Mitsuru　紙山　満

速度の違う機械でも変わらないウエイトをかけるためのプログラム，XVI.FNCです。付加機能としてBASICで垂直帰線期間を知ることもできるので，画面の書き換えなどの際に威力を発揮してくれるでしょう。

スピードが違う!

X68000XVIユーザーの皆さん，プログラムを16MHzで動かしたら変になった，ということはありませんか？　よく市販のゲームなども16MHzで動かすとフェードアウトが終わる前に画面の書き換えを始めてしまったりウエイトが軽すぎてゲームにならなかったりすることがあります。

これはウエイトに空回りを使っているからなのですが，10MHzでちょうどよかったウエイトも16MHzでは空回りが早く終わってしまうために起きる現象なのです（それ以外の原因もあるが)。それでは，垂直同期などの10MHzでも16MHzでも速さが変わらないものでカウントをとればよいのですが，X-BASICでは垂直同期をサポートしていません。

そこで，今回のXVI.FNCという関数を作ってみました。

XVI.FNCとは?

とりあえず，プログラムを入力してみてください。

マシン語入力ツールMAC.Xからリスト2のダンプリストを打ち込んでください(セーブバイト数は554バイト)。または，リスト1のプログラムをエディタから入力し，アセンブル&リンクしてください。生成されたXVI.XをXVI.FNCにリネームします。そしてBASICに組み込んでください。すると，X-BASICでVWAIT()という関数が使えるようになります

この関数は引数によってプログラムに垂直同期を利用したウエイトをかけることができます。この時間は10MHzでも16MHzでも変わりません。

例)

　VWAIT(10)

また，引数を0にすれば単に垂直同期を検出するだけの関数に早変わりします。つまり，

　VWAIT(0)：HOME(0, 0, I)

などとすれば，垂直同期検出機能付きHOME関数になります。こうすれば，スクロールするときにチラチラしたりふにゃふにゃしたりしないようになります。HOMEのほかにも，VPAGE, BG_SCROLLなどにも威力を発揮します。

またXVI.FNCにはXVI()という関数もあります。引数はありません。戻り値には，いま10MHzで動作しているなら－1が，16MHzで動作しているなら0が返ってきます。

これにより，どうしてもこのプログラムは10MHzで動かしてほしいという場合には，プログラムの先頭でXVI()を使い，0なら「10MHzモードにしてください！」などと表示してプログラムを終了させればよいのです。

また，先ほどのVWAIT()だけでは速度を調整しきれないプログラムの場合，16MHz時と10MHz時とでVWAIT()に与える引数を変えてやることもできます。

ライブラリ対応

なお，XVI.SはそのままでCのライブラリにもなっているのです！　ですから，この2つの関数を含むX-BASICのプログラムもコンパイル可能です。

ライブラリを作成する場合はXVI.Sをアセンブルして XVI.Oにしておきます。そしてコンパイルのときに，

　CC　?????.BAS　XVI.O

のようにします。

本当は，ONTIMEによるウエイト関数TWAIT()などもつけたかったのですが，現在でも少し長いプログラムなのにもうひとつ加えるともっと長くなるのでやめました。

実は，私はアセンブラなどあまりわからないのですが，なんとなくカッコイイので，今回はアセンブラでプログラムを組んでみました。そのため，いろいろとムダなコードが多いかもしれませんが，そこはご愛敬ということで。

では，X68000XVIユーザーの方もそうでない方もXVI.FNCを活用してみてください。

リスト1

```
*=====================================
*
*                  XVI.S
*            (BASIC・C共用)
*
*  Programmed 1992 by Mitsuru Kamiyama
*
*=====================================

        .include        doscall.mac
        .include        fdef.h

        .globl  _xvi
        .globl  _vwait

GPIP    equ     $e88001
SPORT   equ     $e8e00b
VDISP   equ     4
CLOCK   equ     0
*---------------------------------------
        dc.l    return,return,return,return
        dc.l    return,return,return,return
        dc.l    ptr_token
        dc.l    ptr_param
        dc.l    ptr_exec
        dc.l    0,0,0,0,0
*---------------------------------------
return:
        rts
*---------------------------------------
ptr_token:
        dc.b    'xvi',0
        dc.b    'vwait',0
        dc.b    0
*---------------------------------------
        .even
ptr_param:
        dc.l    xvi_p
        dc.l    vwait_p
xvi_p:
        dc.w    int_ret
vwait_p:
        dc.w    int_val
        dc.w    void_ret
*---------------------------------------
ptr_exec:
        dc.l    xvi_run
        dc.l    vwait_run
*---------------------------------------
xvi_run:
        bsr     _xvi
        move.l  d0,ret_data
        moveq.l #0,d0
        lea     ret,a0
        rts
_xvi:
        movem.l d1,-(sp)
        clr.l   -(sp)
        DOS     _SUPER
        addq.l  #4,sp
        move.l  d0,-(sp)
        btst.b  #CLOCK,SPORT
        beq     _16
        moveq.l #-1,d1
        bra     xvi_exit
_16:    moveq.l #0,d1
xvi_exit:
        DOS     _SUPER
        addq.l  #4,sp
        move.l  d1,d0
        movem.l (sp)+,d1
        rts
*---------------------------------------
vwait_run:
        move.l  12(sp),d1
        bmi     vwait_err
        move.l  d1,-(sp)
        bsr     _vwait
        addq.l  #4,sp
        moveq.l #0,d0
        rts
_vwait:
        link    a6,#0
        movem.l d0-d1,-(sp)
        clr.l   -(sp)
        DOS     _SUPER
        addq.l  #4,sp
        move.l  8(a6),d1
        beq     vdisp_only
loop1:  btst.b  #VDISP,GPIP
        bne     loop1
loop2:  btst.b  #VDISP,GPIP
        beq     loop2
        subq.l  #1,d1
        beq     vwait_exit
        bra     loop1
vdisp_only:
        btst.b  #VDISP,GPIP
        bne     vdisp_only
vwait_exit:
        move.l  d0,-(sp)
        DOS     _SUPER
        addq.l  #4,sp
        movem.l (sp)+,d0-d1
        unlk    a6
        rts
vwait_err:
        moveq.l #1,d0
        lea     vwait_mes,a1
        rts
*---------------------------------------
ret:
        dc.w    0
        dc.l    0
ret_data:
        dc.l    0
vwait_mes:
        dc.b    '引数が異常です。',0

        .end
```

リスト2

```
0000   1B D6 2D 6C 68 31 2D 08  : 58
0008   01 00 00 9E 01 00 00 00  : A0
0010   60 57 18 20 00 05 58 56  : A2
0018   49 2E 4F 0A 5D 5C C6 62  : B1
0020   61 C6 95 79 3C 5D 5C 29  : 53
0028   98 31 80 9D 28 07 C4 12  : EB
0030   E6 CC BD E6 3A EC 2D 67  : 0F
0038   C3 DB 5E 9E A0 2F 1F 1A  : A2
0040   0B F7 7F E8 6A A3 90 AD  : B3
0048   04 CF B3 43 DF F7 A8 FA  : 41
0050   C0 B6 20 5E BB 79 CF AC  : A3
0058   6C E0 B3 A9 B9 20 74 26  : 1B
0060   9F 84 39 D2 01 BC CC B8  : 6F
0068   16 52 68 05 ED 43 81 7D  : 03
0070   19 CD 6C 92 9E DE 00 8E  : EE
0078   01 64 AC B9 14 93 25 2D  : C3
-----------------------------------
SUM:   71 5C 82 22 61 B4 A4 E5   C45B

0080   F3 B0 2D FC 66 44 31 AB  : 52
0088   ED 65 2B B0 C1 B6 89 81  : AE
0090   70 18 D7 25 03 5F 4F A4  : D9
0098   DA 27 48 8A 8F E3 B9 B5  : B3
00A0   B6 C1 6E 94 22 1E A2 27  : 82
00A8   A9 6B 91 72 5A 9A DD 62  : 4A
00B0   81 BD D9 65 38 A3 85 71  : 4D
00B8   16 8B 6F 9A 4F F3 1F 61  : 6C
00C0   75 41 6C 2A C1 07 B1 81  : 46
00C8   15 2D 0F EB 97 07 31 3D  : 48
00D0   DC 20 12 19 25 EB AF 46  : 2C
00D8   17 FF CA 00 33 B3 7E 88  : CC
00E0   1D A6 6E 81 6D 57 4C 78  : 3A
00E8   FD 58 2C C2 0C C6 D7 71  : 5D
00F0   01 3E E7 B5 7A 09 D4 B5  : E7
00F8   1B 53 94 32 90 92 55 02  : AD
-----------------------------------
SUM:   D3 E4 2A B8 EF EE 40 0C   84D4

0100   4D F0 10 62 13 FF 91 57  : A9
0108   24 CB 2F E0 87 41 BB 8C  : 0D
0110   91 FC A3 E2 30 96 A1 33  : AC
0118   07 A9 3C C9 31 AB 6E AC  : AB
0120   74 E1 96 21 30 1D 81 2D  : 07
0128   6C 68 31 2D E5 00 00 00  : 17
0130   98 01 00 00 00 60 57 18  : 68
0138   20 00 07 58 56 49 2E 46  : 92
0140   4E 43 B6 FC EA 78 71 92  : A8
0148   80 0C 6E 9B 08 19 B0 8C  : F2
0150   01 CA 83 E8 40 1C CA 80  : DC
0158   1F 3A F8 07 DA B4 03 F4  : DD
0160   4E 8F 2E 00 41 00 BE B8  : C2
0168   76 20 7E DB 00 08 F0 87  : 6E
0170   8A D8 0F C8 19 83 43 1D  : 35
0178   17 2F 28 27 E1 28 0C AA  : 54
-----------------------------------
SUM:   F4 B3 6E E3 AD 5B 4C E5   CCBC

0180   BB 39 AF 4C 9C DB C8 7F  : AD
0188   2A 73 6D B3 83 EA 52 6C  : E8
0190   F1 5C EE E0 B9 33 89 AC  : 3C
0198   AA 1B DD C1 94 E2 B4 27  : B4
01A0   6C 9E 65 FE 73 7F E9 3E  : 86
01A8   CA 68 38 70 E1 59 07 3D  : 58
01B0   89 3C 2D 65 B6 D7 47 27  : 52
01B8   31 F7 EE 30 80 DB 24 C6  : 8B
01C0   3E 69 45 59 EA 27 5B 4A  : FB
01C8   50 76 07 6F 1B A3 C6 C4  : 84
01D0   60 28 7E A4 F1 74 E6 7C  : 71
01D8   B0 59 80 4A 06 6E F8 6B  : AA
01E0   27 78 2A 67 F8 32 EB 5C  : A1
01E8   4A AC 93 C7 A7 EA 59 EA  : 24
01F0   63 39 EA 31 D2 42 B0 6B  : E6
01F8   E8 3C F3 BE F4 80 43 AB  : 37
-----------------------------------
SUM:   CA 55 83 76 57 EE E8 77   4BA8

0200   88 D6 73 30 04 A8 BF DC  : 48
0208   15 85 FD 33 32 07 B6 F0  : A9
0210   0E C9 00 3C B7 40 6A 3C  : B0
0218   01 0E F9 5F 09 48 2C C1  : A5
0220   28 22 AF 02 D7 C4 ED 6A  : ED
0228   28 00 00 00 00 00 00 00  : 28
0230   00 00 00 00 00 00 00 00  : 00
0238   00 00 00 00 00 00 00 00  : 00
0240   00 00 00 00 00 00 00 00  : 00
0248   00 00 00 00 00 00 00 00  : 00
0250   00 00 00 00 00 00 00 00  : 00
0258   00 00 00 00 00 00 00 00  : 00
0260   00 00 00 00 00 00 00 00  : 00
0268   00 00 00 00 00 00 00 00  : 00
0270   00 00 00 00 00 00 00 00  : 00
0278   00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:   FC 54 18 00 CD FB F8 33   D28E
```

BASICのスプライト処理高速化
SPRITE2.FNC

Nakamura Fumihiro 中村 文洋

スプライトの書き換えなどをまとめて行えるX-BASIC用の拡張外部関数です。ライブラリにも対応していますので，コンパイルすることでより本格的なゲームを作成することができるようになります。

はじめに

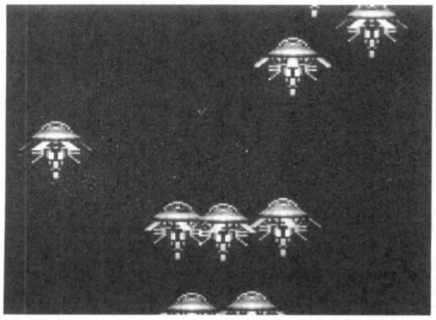
Oh!Xの6月号(1992年)を読んでいてQ&Aのコーナー（158ページ）でスプライトVRAMへのDMA転送に関する質問を見つけました。確かに，大量のパターンを使ってアニメーション処理を行うにはスプライトの書き換えは不可欠です。記事を見て，小泉さんやBASICでゲームを作りたいと思っている人たちのなにかの役に立ってくれればと思い，今回のプログラムを作成しました。

このプログラムは基本的にはX-BASICでのスプライト処理を軽くするための関数群です。たくさんのスプライトパターンをX-BASIC上から使用するために必要なPCGデータやパレットなどの転送処理をまとめてあります。

内部で垂直帰線期間を監視していますので，パターンを書き換えても画面がちらつくこともありません。X-BASICインタプリタ上ではBASICが遅すぎてメリットがあまりわかりませんが，特にコンパイルして使用する場合にはかなりの効果を発揮します。

X68000の場合,同人ソフトなどの多くがX-BASICにより開発されています。基本的な入出力さえ作ってやれば，元がBASICプログラムとは思えないくらいの作品を作ることもできます。特にGCCなどを使用することで，大きな成果を上げることもできます。

X-BASICでメインルーチン，外部関数でサブルーチンを作ることで効率のよいプログラム開発と高いパフォーマンスが得られるのです。今回のSPRITE2.FNCもそのような外部関数の一環となるものといえます。



入力方法

実行形式のプログラムはLHA.Xで圧縮されたかたちで掲載されています（リスト4）。このダンプリストの入力には1992年6月号のMAC.Xを使用します。MAC.Xを起動して“SP2.LZH”などのようにファイル名を指定し，「E」キーでエディットモードに移行してダンプリストを打ち込んでください。入力が終わったら「S」キーでセーブしたあと(セーブ時のバイト数は775バイトに指定する)，LHA.Xで展開してください。これでSPRITE2.FNCとSP2LIB.Oが得られます。

ソースリストから入力する場合は，以下の手順で行ってください。なお，ダンプリストで入力した方は以下の3)以降の手順で続きの作業を行ってください。

なお，サンプルプログラムはグラディウスのキャラクタデータの一部を使用していますので，実行にはグラディウスのディスクが必要です。あらかじめご了承ください。

1) エディタから“sprite2.s”を入力する(リスト1)。

A>ed sprite2.s

2) アセンブル，リンクする。

A>cc sprite2.s

3) できたファイルをX-BASICの外部関数としてBASIC2ディレクトリへコピーする。

A>copy sprite2.x ¥BASIC2¥*.fnc

4) BASIC.CNFへ，

FUNC=sprite2.fnc

のように関数を追加する。

5) エディタから“sp2lib.s”を入力する(リスト2)。

A>ed sp2lib.s

6) アセンブルする。

A>as sp2lib.s

7) BASICを起動する。

8) “sp2test.bas”を入力する(リスト3)。

9) Human68kのコマンドモードへ戻る。

10) Cコンパイラを使って“sp2test.bas”をコンパイルする。

A>cc sp2test.bas sp2lib.o /O

11) グラディウスのディスクから,不可視属性を解除したファイル（ATTRIB -Hを実行),

PALET.PAL

TEKI.CHR

を探し，カレントディレクトリへコピーする。

12) サンプルプログラムを実行する。

A>sp2test.x

このとき，指定されたグラディウスのファイルが見つからないと画面は真っ黒なままですので注意してください。

SPRITE2.FNCの解説

以下にSPRITE2.FNCで拡張される関数を解説します。具体的な使用法については，サンプルプログラムを読んで参考にしてみてください。

●SP_COL2

書式　sp_col2(p, c, pb)

引数　int(p, c, pb)

機能　スプライトのカラーコードを設定する。SP_COLORの代わりに使用するもの。

p……パレットコード（0～15)

c……カラーコード（0～65535)

pb……パレットブロック（0～15)

●SP_PALET

書式　sp_palet(pb，ca，n)

引数　int(pb，n)

char型１次元配列名（ca)

機能　スプライトのパレットブロックを設定する。

pb……パレットブロック（０～15)

ca……パレットデータ配列名

n……データ番号（32×nバイト）

●SP_PCG

書式　sp_pcg(cd，ca，n)

char型１次元配列名（ca)

引数　int(cd，ｎ)

機能　スプライトのパターンを定義する。

cd……パターンコード（０～127)

ca……パターンデータ配列名

n……データ番号(128×nバイト)

●SP_SCRL

書式　sp_scrl(s，x，y，pd，pr)

引数　int(s，x，y，pd，pr)

機能　スプライトのスクロール座標を設定する。

s……プレーン（０～127)

x……Ｘ座標

y……Ｙ座標

pd……パターンデータ(０～&HCFFF)

pr……優先度（０～３)

●SP_OUT

書式　sp_out()

機能　スプライトの表示を行う。

あとがき

プログラム的には，特に高度な技はなにも使ってないので，ソースリストを見ればだいたいなにをやっているのかすぐにわかると思います。

実はこのプログラムは数年前に作ったものの一部なのです。

このプログラムを使用するにあたっての制限は特になにもありませんが，実行する際に，神に感謝することだけは忘れないようにしてください。でないと暴走する可能性があります(嘘ですけど)。では，皆さんSPRITE2.FNCを活用してみてください。

リスト1

```
***
***     Sprite2.fnc
***
***
        .include        iocscall.mac
        .include        fdef.h

IOCS    equ     $0f

SPAL    equ     $e82200

CSR2    equ     $e84080
DCR2    equ     $e84084
OCR2    equ     $e84085
SCR2    equ     $e84086
CCR2    equ     $e84087
MTC2    equ     $e8408a
MAR2    equ     $e8408c
DAR2    equ     $e84094

GPIP    equ     $e88001

SP0     equ     $eb0000
PCG     equ     $eb8000

SUPER_ENT       macro
        moveq.l #_B_SUPER,d0
        movea.l #0,a1
        trap    #IOCS                   * ｽｰﾊﾟｰﾊﾞｲｻﾞﾓｰﾄﾞ
        move.l  d0,ssp_buf
        endm

SUPER_OUT       macro
        moveq.l #_B_SUPER,d0
        move.l  ssp_buf,a1
        trap    #IOCS                   * ﾕｰｻﾞｰﾓｰﾄﾞ
        endm

*========================================
        dc.l    X_INIT
        dc.l    X_RUN
        dc.l    X_END
        dc.l    X_SYS
        dc.l    X_BRK
        dc.l    X_CTRL_D
        dc.l    X_RES1
        dc.l    X_RES2
        dc.l    PTR_TOKEN
        dc.l    PTR_PARAM
        dc.l    PTR_EXEC
        dc.l    0,0,0,0,0
X_INIT:
X_RUN:
X_END:
X_SYS:
X_BRK:
X_CTRL_D:
X_RES1:
X_RES2: rts
*----------------------------------------
PTR_TOKEN:

        .dc.b   'sp_col2',0
        .dc.b   'sp_palet',0
        .dc.b   'sp_pcg',0
        .dc.b   'sp_scrl',0
        .dc.b   'sp_out',0
        .dc.b   0
        .even
*----------------------------------------
PTR_PARAM:

        .dc.l   SP_COL2_PAR
        .dc.l   SP_PALET_PAR
        .dc.l   SP_PCG_PAR
        .dc.l   SP_SCRL_PAR
        .dc.l   SP_OUT_PAR

SP_COL2_PAR:
        .dc.w   int_val
        .dc.w   int_val
        .dc.w   int_val
        .dc.w   void_ret
SP_PALET_PAR:
        .dc.w   int_val
        .dc.w   aryl_c
        .dc.w   int_val
        .dc.w   void_ret
SP_PCG_PAR:
        .dc.w   int_val
        .dc.w   aryl_c
        .dc.w   int_val
        .dc.w   void_ret
SP_SCRL_PAR:
        .dc.w   int_val
        .dc.w   int_val
        .dc.w   int_val
        .dc.w   int_val
        .dc.w   int_val
        .dc.w   void_ret
SP_OUT_PAR:
        .dc.w   void_ret
*----------------------------------------
PTR_EXEC:
        .dc.l   SP_COL2_EXEC
        .dc.l   SP_PALET_EXEC
        .dc.l   SP_PCG_EXEC
        .dc.l   SP_SCRL_EXEC
        .dc.l   SP_OUT_EXEC
*========================================
SUPER_EXEC:
        moveq.l #_B_SUPER,d0
        movea.l #0,a1
        trap    #IOCS                   * ｽｰﾊﾟｰﾊﾞｲｻﾞﾓｰﾄﾞ

        moveq.l #0,d0
        rts
*========================================
SP_COL2_EXEC:
        SUPER_ENT
        move.l  #SP_PALET_WORK,a0
        move.l  12(a7),d0
        lsl.l   #1,d0
        add.l   d0,a0
        move.l  32(a7),d0
        lsl.l   #5,d0
        add.l   d0,a0                   * ワークエリア

        move.l  22(a7),d0               * データ

        move.w  d0,(a0)                 * 移動

        SUPER_OUT
        moveq.l #0,d0
        rts
*========================================
SP_PALET_EXEC:
        SUPER_ENT
        move.l  32(a7),d0
        lsl.l   #5,d0
        move.l  22(a7),d1
        add.l   #10,d1
        add.l   d1,d0
        move.l  d0,MAR2                 * パレットデータ

        move.l  12(a7),d0
        lsl.l   #5,d0
        add.l   #SP_PALET_WORK,d0
        move.l  d0,DAR2                 * ワークエリア

        move.w  #$10,MTC2               * 16ワード

        move.b  #$FF,CSR2
        move.b  #$08,DCR2
        move.b  #$11,OCR2
        move.b  #$05,SCR2
        move.b  #$88,CCR2               * DMA転送

        SUPER_OUT
        moveq.l #0,d0
        rts
*========================================
SP_PCG_EXEC:
        SUPER_ENT
```

```
        move.l  32(a7),d0
        lsl.l   #7,d0
        move.l  22(a7),d1
        add.l   #10,d1
        add.l   d1,d0
        move.l  d0,MAR2                         * PCGデータ

        move.l  12(a7),d0
        lsl.l   #7,d0
        add.l   #PCG,d0
        move.l  d0,DAR2                         * PCGエリア

        move.w  #$40,MTC2                       * 64ワード

        move.b  #$FF,CSR2
        move.b  #$08,DCR2
        move.b  #$11,OCR2
        move.b  #$05,SCR2
        move.b  #$88,CCR2                       * DMA転送

        SUPER_OUT
        moveq.l #0,d0
        rts
*=======================================
SP_SCRL_EXEC:
        SUPER_ENT
        move.l  12(a7),d0
        asl.l   #3,d0
        add.l   #SP_SCRL_WORK,d0
        move.l  d0,a0                           * ワークエリア

        move.l  22(a7),d0
        move.w  d0,(a0)+                        * X座標
        move.l  32(a7),d0
        move.w  d0,(a0)+                        * Y座標
        move.l  42(a7),d0
        move.w  d0,(a0)+                        * VR/HR/COLOR/CODE
        move.l  52(a7),d0
        move.w  d0,(a0)                         * PRW

        SUPER_OUT
        moveq.l #0,d0
        rts
*=======================================
SP_OUT_EXEC:
        SUPER_ENT
        move.l  #SP_PALET_WORK,MAR2             * ワークエリア
        move.l  #SPAL,DAR2                      * Sパレット
        move.w  #$100,MTC2                      * 256ワード

        move.b  #$FF,CSR2
        move.b  #$08,DCR2
        move.b  #$11,OCR2
        move.b  #$05,SCR2

        movea.l #GPIP,a0
SP_OUT_L01:
        btst.b  #4,(a0)
        beq     SP_OUT_L01
SP_OUT_L02:
        btst.b  #4,(a0)
        bne     SP_OUT_L02                      * 垂直帰線期間

        move.b  #$88,CCR2                       * DMA転送

SP_OUT_L03:
        tst.w   MTC2
        bne     SP_OUT_L03                      * 転送終了

        move.l  #SP_SCRL_WORK,MAR2              * ワークエリア
        move.l  #SP0,DAR2                       * ｽﾌﾟﾗｲﾄｽｸﾛｰﾙ
        move.w  #$200,MTC2                      * 512ワード

        move.b  #$88,CCR2                       * DMA転送

        SUPER_OUT
        moveq.l #0,d0
        rts
*=======================================
ssp_buf         .ds.l   1
SP_PALET_WORK   .ds.w   16*16
SP_SCRL_WORK    .ds.w   4*128
```

リスト2

```
***
***     sprite2.fnc用ライブラリ
***
***     SP2LIB.O
***
***
        .include        iocscall.mac

        .globl  _sp_col2
        .globl  _sp_palet
        .globl  _sp_pcg
        .globl  _sp_scrl
        .globl  _sp_out

IOCS    equ     $0f

SPAL    equ     $e82200

CSR2    equ     $e84080
DCR2    equ     $e84084
OCR2    equ     $e84085
SCR2    equ     $e84086
CCR2    equ     $e84087
MTC2    equ     $e8408a
MAR2    equ     $e8408c
DAR2    equ     $e84094

GPIP    equ     $e88001

SP0     equ     $eb0000
PCG     equ     $eb8000

SUPER_ENT       macro
        moveq.l #_B_SUPER,d0
        movea.l #0,a1
        trap    #IOCS                           * ｽｰﾊﾟｰﾊﾞｲｻﾞﾓｰﾄﾞ
        move.l  d0,ssp_buf
        endm

SUPER_OUT       macro
        moveq.l #_B_SUPER,d0
        move.l  ssp_buf,a1
        trap    #IOCS                           * ﾕｰｻﾞｰﾓｰﾄﾞ
        endm

*=======================================
_sp_col2:
        link    a6,#0
        SUPER_ENT
        move.l  #SP_PALET_WORK,a0
        move.l  8(a6),d0
        lsl.l   #1,d0
        add.l   d0,a0
        move.l  16(a6),d0
        lsl.l   #5,d0
        add.l   d0,a0

        move.l  12(a6),d0

        move.w  d0,(a0)
        SUPER_OUT
        unlk    a6
        rts
*=======================================
_sp_palet:
        link    a6,#0

        SUPER_ENT
        move.l  16(a6),d0
        lsl.l   #5,d0
        move.l  12(a6),d1
        add.l   d1,d0
        move.l  d0,MAR2

        move.l  8(a6),d0
        lsl.l   #5,d0
        add.l   #SP_PALET_WORK,d0
        move.l  d0,DAR2

        move.w  #$10,MTC2

        move.b  #$FF,CSR2
        move.b  #$08,DCR2
        move.b  #$11,OCR2
        move.b  #$05,SCR2
        move.b  #$88,CCR2
        SUPER_OUT

        unlk    a6
        rts
*=======================================
_sp_pcg:
        link    a6,#0

        SUPER_ENT
        move.l  16(a6),d0
        lsl.l   #7,d0
        move.l  12(a6),d1
        add.l   d1,d0
        move.l  d0,MAR2

        move.l  8(a6),d0
        lsl.l   #7,d0
        add.l   #PCG,d0
        move.l  d0,DAR2

        move.w  #$40,MTC2

        move.b  #$FF,CSR2
        move.b  #$08,DCR2
        move.b  #$11,OCR2
        move.b  #$05,SCR2
        move.b  #$88,CCR2
        SUPER_OUT

        unlk    a6
        rts
*=======================================
_sp_scrl:
        link    a6,#0

        SUPER_ENT
        move.l  8(a6),d0
        asl.l   #3,d0
        add.l   #SP_SCRL_WORK,d0
        move.l  d0,a0

        move.l  12(a6),d0
        move.w  d0,(a0)+
        move.l  16(a6),d0
        move.w  d0,(a0)+
        move.l  20(a6),d0
        move.w  d0,(a0)+
        move.l  24(a6),d0
        move.w  d0,(a0)
        SUPER_OUT

        unlk    a6
        rts
*=======================================
_sp_out:
        SUPER_ENT
```

```
        move.l  #SP_PALET_WORK,MAR2
        move.l  #SPAL,DAR2
        move.w  #$100,MTC2

        move.b  #$FF,CSR2
        move.b  #$08,DCR2
        move.b  #$11,OCR2
        move.b  #$05,SCR2

        movea.l #GPIP,a0
SP_OUT_L01:
        btst.b  #4,(a0)
        beq     SP_OUT_L01
SP_OUT_L02:
        btst.b  #4,(a0)
        bne     SP_OUT_L02

        move.b  #$88,CCR2

SP_OUT_L03:
        tst.w   MTC2
        bne     SP_OUT_L03

        move.l  #SP_SCRL_WORK,MAR2
        move.l  #SP0,DAR2
        move.w  #$200,MTC2

        move.b  #$88,CCR2
        SUPER_OUT

        rts
*======================================
ssp_buf         .ds.l   1
SP_PALET_WORK   .ds.w   16*16
SP_SCRL_WORK    .ds.w   4*128
```

リスト3

```
10 /*
20 /*       SPRITE2.FNC用サンプルプログラム
30 /*
40 /*          (グラディウスのディスクを用意してください)
50 /*
60 int sx(31),sy(31),xa(31),ya(31),sc(31)
70 int sj(31),sp0(31),sp1(31)
80 char tpal(32-1)={
90          +0,  0,248, 62,255,192,255,254
100        ,206,172,206,172,206,172,206,172
110        ,  9,208,  9,208,  9,208,  9,208
120        ,  9,208,  9,208,  9,208,  9,208}
130 char spal(1120-1),teki(25600-1)
140 /*
150 f=fopen("palet.pal","r")
160 fread(spal,1120,f)
170 fclose(f)
180 /*
190 f=fopen("teki.chr","r")
200 fread(teki,25600,f)
210 fclose(f)
220 /*
230 screen 0,1,1,1
240 sp_init()
250 sp_on()
260 sp_disp(1)
270 sp_palet(0,tpal, 0)
280 sp_palet(1,spal, 0)
290 sp_palet(2,spal, 1)
300 sp_palet(3,spal, 7)
310 sp_palet(4,spal, 8)
320 for i=0 to 6
330     sp_pcg(i,teki,i+65)
340 next
350 /*
360 for n=0 to 15
370     sx(n)=rnd()*256
380     sy(n)=rnd()*256
390     xa(n)=int(rnd()*3)-1
400     ya(n)=int(rnd()*3)-1
410     sc(n)=int(rnd()*4+1)*256
420 sj(n)=rnd()*4
430 next
440 while inkey$(0)=""
450   for n=0 to 15
460     switch sj(n)
470       case 0:sp0(n)=       1:sp1(n)=       2:break
480       case 1:sp0(n)=       5:sp1(n)=       6:break
490       case 2:sp0(n)=       3:sp1(n)=       4:break
500       case 3:sp0(n)=16384+6:sp1(n)=16384+5:break
510     endswitch
520     sj(n)=(sj(n)+1)mod 4
530     sx(n)=sx(n)+xa(n)
540     if sx(n)<-16 then {
550                 sx(n)=272
560                 } else {
570                 if sx(n)>272 then sx(n)=-16
580                 }
590     sy(n)=sy(n)+ya(n)
600     if sy(n)<-16 then {
610                 sy(n)=272
620                 } else {
630                 if sy(n)>272 then sy(n)=-16
640                 }
650     sp_scrl(n*4  ,sx(n)   ,sy(n)   ,sc(n)+sp0(n),2)
660     sp_scrl(n*4+1,sx(n)+16,sy(n)   ,sc(n)+sp1(n),2)
670     sp_scrl(n*4+2,sx(n)   ,sy(n)+16,sc(n)       ,2)
680     sp_scrl(n*4+3,sx(n)+16,sy(n)+16,sc(n)+ 16384,2)
690   next
700   sp_out()
710 /*
720 endwhile
```

リスト4

```
0000  21 3A 2D 6C 68 35 2D 5D  : 1B
0008  01 00 00 DC 02 00 00 16  : F5
0010  11 3D 19 20 01 08 73 70  : 73
0018  32 6C 69 62 2E 6F AE 16  : CA
0020  48 00 00 01 15 5A 76 DA  : 08
0028  36 AA 7F 75 D9 5B 01 07  : 10
0030  B0 D4 1D 01 0A EB 0D 80  : 24
0038  98 CC 26 0A 61 88 C0 BA  : F7
0040  1A 8C 17 15 AE DA 03 43  : A0
0048  53 5B 81 C1 A0 5C 0A B8  : AE
0050  46 23 E0 4A 31 18 16 66  : 58
0058  63 B0 D8 0E 0B 03 45 4C  : 98
0060  50 7F EE EB 98 08 28 20  : 90
0068  C9 60 BE 0B B6 00 4A B6  : A8
0070  E5 C4 00 5B CA 07 EA 3C  : FB
0078  04 66 01 76 21 B1 CD 09  : 89
----------------------------------
SUM:  43 F0 6E 40 B5 E5 23 DC  229B

0080  2D 62 F4 0E 47 21 70 97  : 00
0088  39 0B E3 82 01 1C F4 26  : E0
0090  08 5C 77 94 77 19 0B AF  : B9
0098  03 F5 DD B6 F5 40 A7 EE  : 55
00A0  21 EF B8 F5 88 54 FE F4  : 8B
00A8  BD DB 49 69 B6 10 D8 1D  : 05
00B0  B8 F2 A3 4D 71 0F B7 76  : 47
00B8  10 7D 26 F7 C0 7F 05 2B  : 19
00C0  E8 A9 B7 59 A8 F7 4B DE  : 69
00C8  B1 CE 66 6F A5 2C 1E 8B  : CE
00D0  3B 21 7F 3C 3E 31 91 9E  : B5
00D8  45 1E 8D 38 8A 4E F4 D1  : C5
00E0  7A 8D D6 6F 89 BB BF 94  : E3
00E8  9A 2A 97 A4 52 B8 B0 35  : EE
00F0  18 C7 57 B1 15 19 30 E2  : 27
00F8  F0 9A C3 B6 9F 2F 6E D4  : 13
----------------------------------
SUM:  4C C5 A5 32 C7 E5 A3 63  3E08

0100  A0 E4 7D 7A 72 8F 62 38  : 16
0108  88 D7 23 E7 23 A0 47 D0  : 43
0110  47 9C 47 D2 47 84 8F AB  : 01
0118  FB 01 E0 46 03 93 EB 01  : A4
0120  C8 58 08 92 21 EF 7D AB  : F2
0128  FF 7D D0 45 AF D2 25 6D  : A4
0130  B3 D8 3E B3 35 53 C8 29  : F5
0138  E4 F4 8B 09 1D 5F 8E 53  : C9
0140  A0 AA 9F DF 4C B5 2C DC  : D1
0148  6F C2 8D 25 A9 46 3A 41  : 4D
0150  5A 09 87 EE 82 CD F1 8E  : A6
0158  23 2E 71 3E D6 54 0D 9C  : D3
0160  B9 B5 3D EB C8 9F B3 EA  : 9A
0168  AB 09 65 7A F3 58 64 13  : 55
0170  58 56 CD 61 F9 E0 39 C5  : B3
0178  97 DD A6 50 D9 E5 F5 48  : 65
----------------------------------
SUM:  A7 8D A1 52 DB 91 C4 99  CC45

0180  24 C6 2D 6C 68 35 2D 60  : AD
0188  01 00 00 60 09 00 00 70  : DA
0190  10 3D 19 20 01 0B 73 70  : 75
0198  72 69 74 65 32 2E 66 6E  : E8
01A0  63 3E E3 48 00 00 01 0C  : D9
01A8  5A 76 EE D1 2F F7 FE E3  : 96
01B0  A4 E2 1C 0C 40 E4 8E 39  : 99
01B8  8B 03 86 8A 73 47 0E 24  : 8A
01C0  B0 56 2B 2D 14 92 84 20  : A8
01C8  F1 2A 12 11 1C 85 37 11  : 27
01D0  16 28 B4 45 E0 1B 81 C5  : 78
01D8  66 68 B0 39 EC 12 1D 81  : 53
01E0  B2 42 70 77 EF FF 37 74  : 74
01E8  E0 B0 30 39 28 88 BC 12  : 77
01F0  BC 9E FA 9D 6A 56 E4 9D  : 32
01F8  B2 58 92 F7 71 64 9C 5F  : 63
----------------------------------
SUM:  B0 FD FA 00 74 15 6D F3  76BC

0200  C9 AF A7 9A 1F 7C 3F 6F  : 02
0208  CC E6 BD 6A 8D ED CF A7  : C9
0210  4B 4C 81 B7 B9 4B 7A AA  : F7
0218  30 4F A2 65 52 7F 05 23  : 7F
0220  3A 75 AA A2 33 8C 3D 90  : 87
0228  FD 21 FA C3 F6 12 A7 E8  : 72
0230  20 03 33 E0 D5 87 F6 32  : BA
0238  91 77 11 28 09 4E 10 FE  : A6
0240  84 4A CC 6E C3 3C 48 BF  : 0E
0248  EC 36 34 1F BE 59 56 C7  : A9
0250  F7 C5 81 95 5B AC 68 94  : D5
0258  7F 27 7B ED 83 61 EC 27  : 05
0260  27 49 C1 17 5C E0 F2 4C  : C2
0268  A2 2D 91 13 E1 68 57 B2  : C5
0270  EE 58 1F 7B CD 74 1D 2D  : 6B
0278  E7 6E 5A 3A BA 3E 3D 5A  : 78
----------------------------------
SUM:  7C E8 36 7B E1 42 0C 51  E3E0

0280  8F 94 80 7C F9 AF 94 00  : 5B
0288  DC 06 B8 3E 40 79 80 F9  : 0A
0290  41 E5 03 B2 0D D0 7C DF  : 13
0298  97 7B F9 2E F7 CE EE F3  : DF
02A0  85 15 DD AF FB BC 7F 37  : 93
02A8  0F 6B AD 75 86 91 C4 28  : 9F
02B0  E2 F4 0C B4 0C CF BE 33  : 62
02B8  94 F9 F0 A9 EA D1 86 F3  : 5A
02C0  25 87 52 A8 5F BA 94 75  : C8
02C8  4D E7 14 5E 41 2D 47 01  : 5C
02D0  8D 0C 10 EB 7D 6A E7 3F  : A1
02D8  43 D4 EB AB 43 5D 78 BD  : 82
02E0  AB AA C3 5D C7 B8 C9 29  : E6
02E8  29 29 29 29 2B 0A 5B 1F  : 53
02F0  A6 C5 86 5C 58 73 8A 30  : D2
02F8  79 25 9E 5A A0 77 43 C7  : B7
----------------------------------
SUM:  82 72 2B F3 FE 0D 30 01  09D1

0300  26 40 C6 2D A2 D0 00 00  : CB
0308  00 00 00 00 00 00 00 00  : 00
0310  00 00 00 00 00 00 00 00  : 00
0318  00 00 00 00 00 00 00 00  : 00
0320  00 00 00 00 00 00 00 00  : 00
0328  00 00 00 00 00 00 00 00  : 00
0330  00 00 00 00 00 00 00 00  : 00
0338  00 00 00 00 00 00 00 00  : 00
0340  00 00 00 00 00 00 00 00  : 00
0348  00 00 00 00 00 00 00 00  : 00
0350  00 00 00 00 00 00 00 00  : 00
0358  00 00 00 00 00 00 00 00  : 00
0360  00 00 00 00 00 00 00 00  : 00
0368  00 00 00 00 00 00 00 00  : 00
0370  00 00 00 00 00 00 00 00  : 00
0378  00 00 00 00 00 00 00 00  : 00
----------------------------------
SUM:  26 40 C6 2D A2 D0 00 00  1721
```

グラフィック画面の3D回転

3DRT_256

Watanabe Takao 渡辺 貴生

最後はグラフィックを3次元的パースをつけて回転させるルーチンです。うまく使えばF-ZERO風の画面効果を持ったゲームも作成可能。ゲームだけでなく応用範囲は広いプログラムといえます。

回転拡大縮小といえばZOOMROT, XROT0.Xなど，たくさんのプログラムがありますが，スーパーファミコンのF-ZEROみたいに3次元にパースをつけて回転してくれるものはなかなかありません。誰かが作ってくれないかと思っていましたが，その気配がないので，自分でこのプログラムを作ってみました。

今回は元グラフィックが256×256ドット，256色，拡大縮小倍率は64段階の高速バージョンを発表させていただきます。このプログラムのほかに，PICデータを読み取り，元グラフィックが512×512ドット，65536色で回転する中速バージョン。元グラフィックが256×2048ドットで，その上を走るカーレースゲームバージョン(ゲームは，未完成)があります。

皆さんがいちばん注目しているのは，やっぱりスピードのはずです。実際に試してみないとわからないと思いますが，スピードは，十分満足できる速さに仕上がったと思います。

スピードを速くするのために仕様が，

1) 元グラフィックの大きさが256×256ドット
2) 表示が256色 (SM.Xを使うので16色)
3) 表示画面が128×128ドット (一度に4ドット描いている)

になりました。ですから，自分でグラフィック描きたいときは，横内威至さんの至高のスプライトメーカーSM.Xを使ってください(感謝，感謝)。後ろに，詳しいデータの作り方を書いてあります。

遊び方

とりあえずサンプルプログラムを見てもらうのがもっともわかりやすいでしょう。リスト2をマシン語入力ツールで入力し(セーブバイト数4592) LHA.Xで展開してください。サンプルの実行に必要なファイルがすべて展開されます。あとは，“3DRT 256.X”を実行するだでけです。

もし，サインテーブルデータ，スプライトデータ，パレットデータのデータのどれかを忘れて実行しても，「○○データがありません」みたいに親切に教えてくれません。ただ，無言で空のデータを呼び出しまくっています。そのときは，素直にESCを押して脱出しましょう。以下が，必要なファイルです。

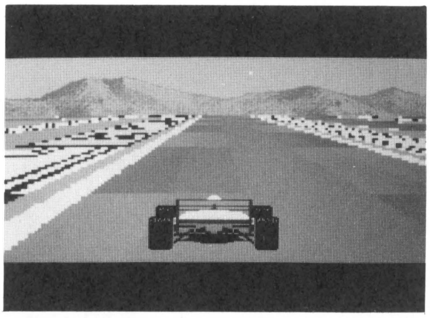
“3DRT_256.X”
“sintbl.dat”
“3DRT_256.SP”
“3DRT_256.PAL”

注：最初に3DRT_256を動かすときは，初期化するらしいので，2，3秒ほど，割り込みが入ってくれません。画面チェンジがおかしいのは，そのせいだと思ってください (バグかもしれない)。

操作方法は以下のとおり。ジョイスティック1とESCを使います。

拡大縮小倍率	64段階
角度	128段階
速度	128段階

```
        拡大
         ↑
右←     ○   →左   トリガ1   トリガ2
         ↓          ○         ○
        縮小        減速      加速
    ESC  脱出
```


レースゲーム風アレンジ例

*　　*　　*

このプログラムを使えばスーパーファミコンのF-ZEROのように地面用に描かれたグラフィックのなかを走り回ることができるのですが，スーパーファミコンとX68000の回転拡大縮小の方法は基本的に違います。

スーパーファミコンは，バックグラウンド (BGのことです) を使う方法です。これはハードウェアでサポートされており，1ラインごとに拡大縮小倍率，位置，角度を決められるので，画面の上下で違う場面を出すことができます。たとえば，スーパーマリオカートがそうです。

X68000は一生懸命計算しまくる方法です。グラフィック画面を使い，1ドットずつ，一定の値を足したり，引いたりして，色を調べて，VRAMに書く方法です。まずアルゴリズムを解説しましょう。

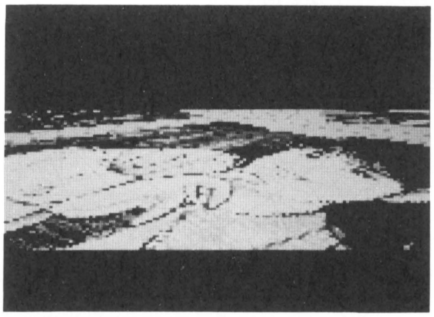
アルゴリズムについて

基本的には，回転拡大縮小の方法と同じですが，XROT0.Xのように1ライン分のデータを配列にする方法はできないので，1ドットずつ計算していくことになります。

まず基本知識として，データ表現を解説します。このプログラムで扱うデータは，

```
  #$00_00_00.00
   (整数)     (小数点以下)
```

のように下位8ビットが小数点以下となっ


512×512ドット版

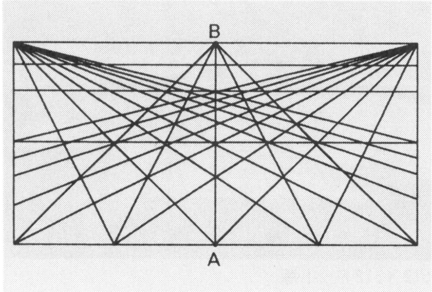
ています。覚えておいてください。

回転の際の三角関数計算にはサインテーブルを使います。サインテーブルはひとつの角度を2バイトで表します。64倍（小数点以下を入れると#$4000倍）されていて，128方向あります。1単位方向を変えるには，2バイト単位なので2を足します。

では，サンプルプログラムの動作を簡単に解説します。

1)　メインループでkakudo，size，x_zahyou，y_zahyou，pic_adrsを設定する
2)　3DRTのサブルーチンに飛ぶ
3)　メインループで設定されたkakudoとsizeを取り出して，basicデータを作る。このデータが基本になる
4)　縦ループでdivsを使うときにオーバーフローを起こさないためにbasicデータを4で割る
5)　basicデータを64分の1＋256分の1したstartデータを作る。256分の1を足すのはグラフィックの回転位置を上げるため(足さないと回転位置が最下段になる)
6)　X，Y座標をstartデータに足す
7)　縦カウントをD5にpic_adrsをA0にwdataをA1に設定する
8)　グラフィック表示位置と割り込み許可を設定する
9)　縦ループにいく
10)　basicデータをD0にコピーしてD5（縦カウント）で割る
11)　D0を符号拡張する
12)　割り終わったのでbasicデータに4を掛ける
13)　startデータをD2にコピーしてD0で引く。D2が画面の縦の中心線の座標となる
14)　D2をw_strデータにコピーする
15)　D0を64分の1にする。この値が横移動量となる
16)　右半分を描く
17)　D2(縦の中心線の座標)から右端まで描く
18)　描き終わったらw_strデータをD2に入れ直す
19)　左半分を描く

図1



20)　D2(縦の中心線の座標)から左端まで描く
21)　終了条件があうまで（D5（縦カウント）が260なら終了）縦ループに飛ぶ
22)　3DRTのサブルーチンを終わる
23)　ESCが押されていなかったらメインループへ，押されていたら終了

これが3DRTの簡単（かな？）な流れです。このように，basicデータの値がstartデータ，横移動量，縦の距離のデータを決めているので，とても理解しにくいアルゴリズムになっています（作った本人にも理解しにくい……)。

役に立つかわかりませんが，横移動量，縦の距離の変化の関係をグラフと図にしておきました（図1，2，3)。

図1は表示する3次元の世界を線で表したものです。私は，この図を元にA－B線からの横移動量，A点からの距離を導きました。

図2は，図1のA－B線からの横移動量，

$$(D0) = \frac{basic \times 64}{縦カウント \times 4}$$

のグラフです。D0とD1は同じ処理をするのでD1のほうは省略しました。

図3は，図1のA点からの縦の距離，

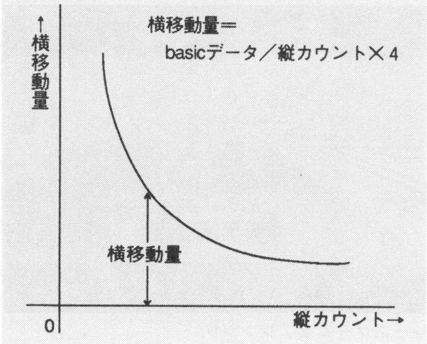
$$(D2) = startデータ - \frac{basicデータ}{縦カウント \times 4}$$

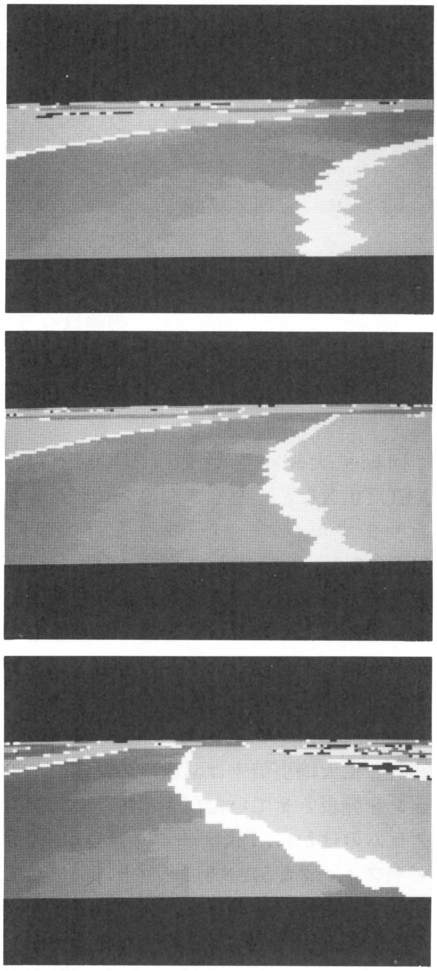
のグラフです。

垂直帰線期間処理

めんどくさかったです。垂直同期を見ないで画面チェンジをしても，ちらつかなかったので，「ま，いいかぁー」とごまかそうとしましたが，バックグラウンドとスプライトを使うときにやっぱり必要になったので，しかたなくとることにしました。スピードを遅くしないためにVDISPSTを使い，垂直帰線期間ごとに割り込みを入れること

図2



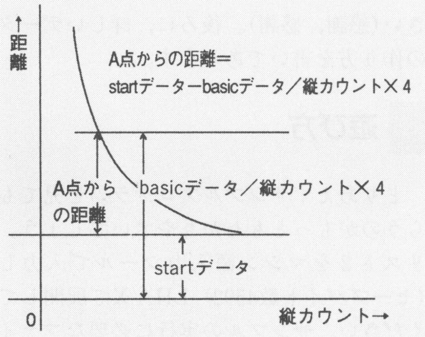


回転の様子（3 DRT_256）

にしました。

方法は，まず，左上にグラフィックを描きます。描き終わったら，左上の表示許可を出します。左下にグラフィックを描きます。描き終わったら，左下の表示許可を出します。右上にグラフィックを描きます。描き終わったら，右上の表示許可を出します。左上にグラフィックを描きます。描き終わったら，左上の表示許可を出します。このように繰り返しているあいだ，常に割り込みが入り，許可情報をチェックします。許可情報はgfflgに入っています。1バイト

図3



です（X000 0XXX）。

　第０ビット　左上の表示許可
　第１ビット　左下の表示許可
　第２ビット　右下の表示許可
　第７ビット　第０～２ビットの許可が入ると，同時に許可

　このほかのビットは使用しません。割り込みが入ったときまず最初に第７ビットをチェックします。許可が出ていなければ割り込みを終了します。もし許可が出ていれば，第０～２ビットをそれぞれチェックして表示位置をスクロールさせ，割り込みを終了します。

　画面チェンジの様子はCRTMODを８にすればわかります。

3DRTを使うには

　3DRTはサブルーチンのかたちで提供されています。使用するプログラムに3DRTの部分をつけ足し，そのプログラムのメインループでkakudo，size，x_zahyou，y_zahyou，pic_adrsの各ワークエリアに値を設定してください。あとは3DRTを呼べば，勝手にそのデータを取り出しグラフィックを描きます。

●kakudo

　１バイトで表し，360度を128方向にしています。１段階変えるには，２を足してください。値を奇数にして3DRTを呼ぶとエラーが起こります。

例：kakudo＝64→90度

●size

　１バイトで表し，64倍率あります。値は０～63です。64以上で3DRTを呼んでもANDされて63以下になります。

例：size＝65→１

●x_zahyou，y_zahyou

　４バイトで表します。下位８ビットが小数点以下になっています。４バイトに格納しますが，使用するデータは下位２バイトです。

例：0013FFFF→0000FFFF

●pic_adrs

　表示させる絵（pic）をpic_adrsに格納します。格納されたアドレスが表示するpicとなります。

例：

```
    lea.l      pic,a0
    move.l     a0,pic_adrs
```

　SM.Xで使用しているSPデータは16×16ドットのパターンで並んでいて，なおかつ，１バイトに２ドット分のデータが入っています。サンプルではこのSPデータ（32768バイト）を縦横256×256ドットの3DRT用picデータに変換しています。

　3DRT用picデータは65536バイトで縦256×横256ドットを表し，１バイトでひとつの色を格納します。1～256バイトが最上段のラインの０～255ドット。257～512バイトがその下のラインの０～255ドットという具合に対応しています。

　このように配置されたデータを指定すれば3DRTルーチンはそのグラフィックを回転させることができます。

　ここで書かれているメインループはサンプルプログラムなので，いろいろと改造してください。

　改造のポイントとしては，

1）　startデータの値を適当に変えてみる（このデータを変えてもあとに影響はない）。実行すると，回転軸の位置が移動する

2）　3DRTの最終行の，add #4を#1にして，縦カウントの値を64までにする。実行すると回転軸の位置が移動し横幅が狭くなる

注：縦カウントの値を７以下にするとdivsでオーバーフローを起こしてグラフィックの最上段がおかしくなる場合（sizeの値が大きい場合）がありますが，エラーは起こりません。

高速化について

　私はTONBEさんのZOOMROTを見て，X68000でもこんなに速く拡大回転縮小できるものかと，もの凄く感動しました。自分も今回のプログラムを作ってみようと思ってソースリストを見たのですが，全然わからないので自分なりの方法で作ることにしました。

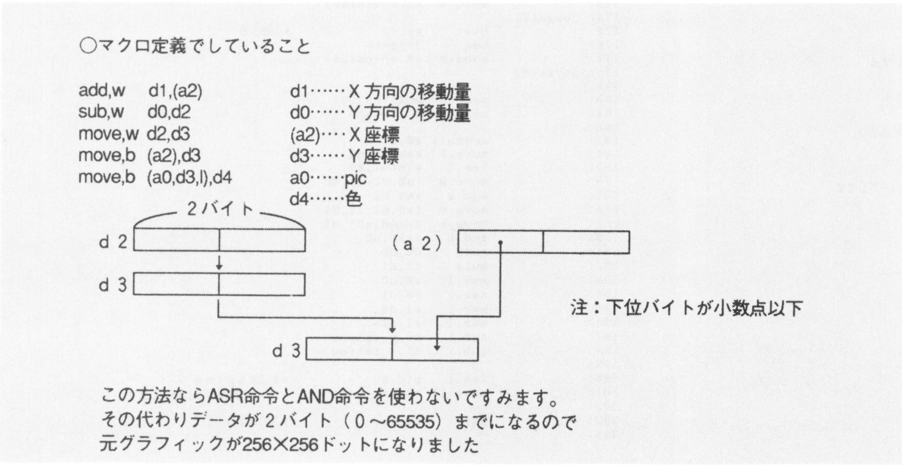
　最初はＣ言語で書きました。苦労の末やっとバグをとって完成。しかし，実行してみたら，「遅い」。そのときは，画面切り換えもしていなかったので，上から下へ書き換える様子がゆっくり見えてしまいました。なぜこんなに遅いのかと，実行ファイルを逆アセンブルして肝心の表示部分を見てみると「なんでこんなことを……」と思うくらい余計なことをしていました。そのとき，私はMPUと直接お話ができるようにアセンブラの勉強する決心をしたのです（決心というほど大げさでないが……）。

　村田敏幸さんの『X68000マシン語プログラミング入門編』を買ってきて，勉強してみると「なんだー」と，実はマシン語は簡単で（Ｃ言語を知っていたからでもある），これまで自分は食わず嫌いをしていただけだとわかりました。

　やっと，アセンブラで書き直して，バグをとって，完成。Ｘファイルを実行してみると，さすがに少しは，速くなりました。え，「少しは……」というわけで，思っていたより速くなりませんでした。GCCって凄いですね。

　しかし，Ｃ言語にはできない命令として，たとえば，swap命令（レジスタの前と後ろ

図４　縦横が256になったわけ



自分でグラフィックを描く

　サンプルのマップ以外のデータを使用したときには以下の手順で作業を行います。

1）　Oh！X６月号のSM.Xを立ち上げて絵を描く

2）　描き終わったらCANVASの絵をスプライトにすべて移す

3）　スプライトデータを”3DRT_256.SP”のファイル名ですべてのデータをセーブ

4）　パレットデータを”3DRT_256.PAL”のファイル名でひとつセーブ

5）　SM.Xを終了

6）”3DRT_256.SP”，”3DRT_256.PAL”のファイルを”sintbl.dat”と”3DRT_256.x”があるディレクトリに置く。そのときに最初からあるスプライトデータとパレットデータのファイル名を変更または，削除する

7）　”3DRT_256.x”を実行

を交換します），D0レジスタの下位８ビットだけをD1レジスタの下位８ビットに転送するmove命令（Cでは，ここまで細かく書けない）などがあります。これらの命令を使って，また書き直してみたら，なんと「速い！」。

というわけで，話が長くなりましたが，スピードを追求するプログラムは，

「やっぱりアセンブラだねっ」

という結論です。

最後に

このプログラムをX68000XVIの16MHzで動かしてみると，感覚では２倍のスピードで動いているように見えました。このスピードなら，XVIでF-ZEROのようなゲームを作ることができると思います。

というわけで，3Dの拡大，縮小，回転の方法は確立できたと思います。バグはないと思いますが，このプログラムによる損害が出ても，いっさい受け付けないことにします。最後にSM.Xの作者，横内威至さんに感謝します。ありがとうございました。

参考文献

C compilerPRO-68Kマニュアル

「X68000マシン語プログラミング入門編」，村田敏幸，ソフトバンク

「Inside X68000」，桑野雅彦，ソフトバンク

68000 PROGRAMMER'S HAND BOOK，技術評論社

リスト1

```
    .include        doscall.mac
    .include        iocscall.mac

                *マクロ定義
dotputr         macro   param               *右ドットプットマクロ定義
                add.w   d1,(a2)             *(A2)=(A2)+D1
                sub.w   d0,d2               *D2=D2-D0
                move.w  d2,d3               *D3=D2
                move.b  (a2),d3             *D3の下位8ビット=(A2)
                move.b  (a0,d3.l),d4        *D4=(A0+D3)：2次元配列
                move.w  d4,(a3)+            *(A3)=D4
                move.w  d4,(a3)+            *A3はグラフィックを打つアドレス
                move.w  d4,(a4)+            *(A4)=D4
                move.w  d4,(a4)+            *A4はグラフィックを打つアドレス
                endm

dotputl         macro   param               *左ドットプットマクロ定義
                sub.w   d1,(a2)
                add.w   d0,d2
                move.w  d2,d3
                move.b  (a2),d3
                move.b  (a0,d3.l),d4
                move.w  d4,-(a3)
                move.w  d4,-(a3)
                move.w  d4,-(a4)
                move.w  d4,-(a4)
                endm

                .text
                .even

                lea.l   worktop(pc),a6
start:
                lea.l   mysp,sp
                IOCS    _OS_CUROF           *カーソル表示をOFFにします。
                move.w  #10,d1
                IOCS    _CRTMOD             *CRTのモードを指定します。
                IOCS    _G_CLR_ON           *グラフィック画面をクリアして表示モードにします。

                clr.l   -(sp)
                DOS     _SUPER              *スーパーバイザモードへ
                move.l  d0,(sp)

                clr.w   -(sp)
                pea.l   palfile
                DOS     _OPEN               *PALファイルをオープンする
                addq.l  #6,sp
                move.l  #32,-(sp)
                pea.l   w_pic
                move.w  d0,-(sp)
                DOS     _READ               *ファイルからデータを読み込む
                lea.l   10(sp),sp

                lea.l   w_pic,a0            *パレットを格納
                move.l  #15,d0
                move.l  #$E82000,a1
stpal:                  move.w  (a0)+,(a1)+
                dbra    d0,stpal

                clr.w   -(sp)
                pea.l   sintblfile
                DOS     _OPEN               *sintblファイルをオープンする
                addq.l  #6,sp
                move.l  #256,-(sp)
                pea.l   sintbl
                move.w  d0,-(sp)
                DOS     _READ               *ファイルからデータを読み込む
                lea.l   10(sp),sp

                clr.w   -(sp)
                pea.l   spfile
                DOS     _OPEN               *SPファイルをオープンする
                addq.l  #6,sp
                move.l  #32768,-(sp)
                pea.l   w_pic
                move.w  d0,-(sp)
                DOS     _READ               *ファイルからデータを読み込む
                lea.l   10(sp),sp

                DOS      ALLCLOSE           *すべてのファイルをクローズします

                *SPデータを3DRT用データに変換
                lea.l   w_pic,a0
                lea.l   pic,a1
                moveq.l #0,d0

                move.l  #16-1,d4
henkan3:        move.l  #32-1,d3
henkan2:        move.l  #16-1,d2
henkan1:        move.l  #4-1,d1
henkan0:
                move.b  (a0),d0
                lsr.b   #4,d0
                move.b  d0,(a1)+
                move.b  (a0)+,d0
                and.b   #$F,d0
                move.b  d0,(a1)+

                dbra    d1,henkan0
                adda.l  #256-8,a1
                dbra    d2,henkan1
                suba.l  #256*16-8,a1
                dbra    d3,henkan2
                adda.l  #256*15,a1
                dbra    d4,henkan3

                *変換終了

                *初期設定
                move.b  #0,gfflg            *グラフィックFLG
                move.b  #15,size            *倍率（0−63段階）
                move.b  #0,kakudo           *角度
                move.l  #0,x_zahyou         *初期表示位置　X座標
                move.l  #0,y_zahyou         *初期表示位置　Y座標

                move.b  #0,muki(a6)         *向き
                move.b  #0,speed(a6)        *スピード

                moveq.l #1,d1               *垂直帰線期間で割り込み設定
                lea.l   gfsd1(pc),a1
                IOCS    _VDISPST

                moveq.l #0,d1
                moveq.l #0,d2
                moveq.l #0,d3
                moveq.l #0,d4

                *mainloop
mainloop:
                moveq.l #0,d1
                IOCS    _JOYGET             *ジョイスティックデータ読み込み
                not.b   d0                  *ジョイスティックデータ反転

                btst    #1,d0               *下
                beq     joyget1
                addq.b  #1,size
joyget1:
                btst    #0,d0               *上
                beq     joyget2
                subq.b  #1,size
joyget2:
                btst    #3,d0               *右
                beq     joyget3
                move.b  muki(a6),d1
                cmp.b   #-6,d1
                beq     joyget6
                subq.b  #2,muki(a6)
                jmp     joyget6
joyget3:
                btst    #2,d0               *左
                beq     joyget4
                move.b  muki(a6),d1
                cmp.b   #6,d1
                beq     joyget6
                addq.b  #2,muki(a6)
                jmp     joyget6
joyget4:
                tst.b   muki(a6)
                beq     joyget6
                bgt     joyget5
                addq.b  #2,muki(a6)
                jmp     joyget6
joyget5:
                subq.b  #2,muki(a6)
joyget6:
                btst    #5,d0               *ボタンA
                beq     joyget7
                addq.b  #1,speed(a6)
joyget7:
                btst    #6,d0               *ボタンB
                beq     joyget8
                subq.b  #8,speed(a6)
joyget8:

                move.b  muki(a6),d1
                add.b   d1,kakudo

                moveq.l #0,d2
                move.b  kakudo,d2
                lea.l   sintbl,a0
                move.w  (a0,d2.l),d0
                add.b   #64,d2
                move.w  (a0,d2.l),d1
                move.b  speed(a6),d2
                and.b   #$7F,d2
                muls    d2,d0
                muls    d2,d1
                asr.l   #8,d0
                asr.l   #8,d1
                asr.l   #1,d0
                asr.l   #1,d1
                sub.l   d0,x_zahyou
                sub.l   d1,y_zahyou

                lea.l   pic,a0              *表示するpicを
                move.l  a0,pic_adrs         *pic_adrsに格納します。

                jsr     TDRT                *3DRTへ!!!
```

```
		moveq.l #0,d1
		IOCS	_BITSNS		*ESCキーの押下状態のセンス
		btst	#1,d0
		beq	mainloop	*mainloopへ

		*終了処理
		move.l	#0,a1
		IOCS	_VDISPST	*垂直同期による割込み禁止
		DOS	_SUPER		*ユーザーモードへ
		addq.l	#4,sp
		IOCS	_OS_CURON	*カーソル表示をONにします。
		move.w	#16,d1
		IOCS	_CRTMOD		*CRTのモードを指定します。
		DOS	_EXIT		*終了

*ここからが3DRT!!!!!です。
*3DRTのアルゴリズムを見ながら読んで下さい。

TDRT:
		*basicデータを作ります
		moveq.l #0,d2
		move.b	kakudo,d2	*D2=角度
		lea.l	sintbl,a0
		move.w	(a0,d2.l),d0	*sin=sin角度
		add.b	#64,d2		*角度=角度+90度
		move.w	(a0,d2.l),d1	*cos=cos角度
		move.b	size,d2		*D2=size
		and.b	#$3F,d2		*D2=D2/64の余り
		muls	d2,d0		*D0=sin*size
		muls	d2,d1		*D1=cos*size
		asr.l	#2,d0		*オーバーフロー防止のため4で割る
		asr.l	#2,d1		*オーバーフロー防止のため4で割る
		move.l	d0,basic_x	*basic_x=d0
		move.l	d1,basic_y	*basic_y=d1

		*startデータを作ります
		asr.l	#6,d0		*D0=D0/64
		asr.l	#6,d1		*D1=D1/64
		move.l	d0,d3		*D3=D0
		move.l	d1,d4		*D4=D1
		asr.l	#2,d3		*D3=D3/4
		asr.l	#2,d4		*D4=D4/4
		add.l	d3,d0		*D0=D0+D3
		add.l	d4,d1		*D1=D1+D4
		add.l	x_zahyou,d0	*D0=D0+X座標
		add.l	y_zahyou,d1	*D1=D1+Y座標
		move.l	d0,start_x	*start_x=D0
		move.l	d1,start_y	*start_y=D1

		moveq.l #0,d2
		moveq.l #0,d3
		moveq.l #0,d4

		move.l	pic_adrs,a0	*A0=pic_adrs
		move.l	#8,d5		*D5=縦カウント
		lea.l	w_data,a2	*A2=w_data

		*グラフィックを表示する場所を調べます
		btst.b	#0,gfflg
		bne	migi
		btst.b	#2,gfflg
		bne	sita

		move.b	#%1000_0100,gfflg *右上のスクロール許可を出します
		move.l	#$C18100,a3	*左上に書きます
		move.l	#$C18500,a4
		jmp	tateloop
sita:
		move.b	#%1000_0001,gfflg *左上のスクロール許可を出します
		move.l	#$C58100,a3	*左下に書きます
		move.l	#$C58500,a4
		jmp	tateloop
migi:
		move.b	#%1000_0010,gfflg *左下のスクロール許可を出します
		move.l	#$C18300,a3	*右上に書きます
		move.l	#$C18700,a4
tateloop:
		*横移動量(D0、D1)と縦の距離(D2)を作ります
		move.l	basic_x,d0
		move.l	basic_y,d1

		divs	d5,d0		*basic/D5
		divs	d5,d1		*basic/D5

		ext.l	d0		*符号拡張
		ext.l	d1		*符号拡張

		asl.l	#2,d0		*割り終ったので4を掛けます
		asl.l	#2,d1		*割り終ったので4を掛けます

		move.l	start_x,d2	*D2=start_x
		sub.l	d0,d2		*D2=D2-D0
		move.w	d2,w_str_x	*w_str_x=D2
		move.w	d2,(a2)		*(A2)=D2

		move.l	start_y,d2	*D2=start_x
		sub.l	d1,d2		*D2=D2-D1
		move.w	d2,w_str_y	*w_str_y=D2

		asr.l	#6,d0		*D0=D0/64
		asr.l	#6,d1		*D1=D1/64

migiloop:
		*グラフィックを書きます
		move.w	d2,d3
		move.b	(a2),d3
		move.b	(a0,d3.l),d4
		move.w	d4,(a3)+
		move.w	d4,(a3)+
		move.w	d4,(a4)+
		move.w	d4,(a4)+

		dotputr d1	*マクロ定義を見てください

		↑以下, 62回展開 (計63個)


		sub.l	#$100,a3	*左へ128ドット
		sub.l	#$100,a4	*左へ128ドット

		*縦の距離を入れ直します
		move.w	w_str_x,(a2)	*(A2)=w_str_x
		move.w	w_str_y,d2	*D2=w_str_y

hidariloop:
		dotputl d1

		↑以下, 63回展開 (計64個)

		add.l	#$900,a3	*右へ128ドット、下へ2ドット
		add.l	#$900,a4	*右へ128ドット、下へ2ドット

		addq.w	#4,d5		*D5=D5+4
		cmp.w	#260,d5		*D5(縦カウント)=260なら終了
		bne	tateloop	*縦ループへ

		rts			*3DRT終了

gfsd1:		bclr.b	#7,gfflg		*垂直帰線期間の割り込み
		beq	retn

		btst.b	#0,gfflg
		beq	gfsd2
		move.l	#$0000_0000,$E80018	*左上にスクロールします
		move.l	#$0000_0000,$E8001C
		jmp	retn
gfsd2:
		btst.b	#1,gfflg
		beq	gfsd3
		move.l	#$0000_0100,$E80018	*左下にスクロールします
		move.l	#$0000_0100,$E8001C
		jmp	retn
gfsd3:
		move.l	#$0100_0000,$E80018	*右上にスクロールします
		move.l	#$0100_0000,$E8001C

retn:		rte

		.even
		.offset 0

muki:		.ds.b	1
speed:		.ds.b	1
worklen

*		ワークエリア
*3DRTで使用するデータです
		.bss
		.even
worktop		.ds.b	worklen
		.even
pic:		.ds.b	65536
w_pic:		.ds.b	32768
sintbl:		.ds.w	128
pic_adrs:	.ds.l	1
w_data:		.ds.w	1
x_zahyou:	.ds.l	1
y_zahyou:	.ds.l	1
basic_x:	.ds.l	1
basic_y:	.ds.l	1
start_x:	.ds.l	1
start_y:	.ds.l	1
w_str_x:	.ds.w	1
w_str_y:	.ds.w	1
gfflg:		.ds.b	1
size:		.ds.b	1
kakudo:		.ds.b	1

*		データ
		.data
		.even
palfile:	.dc.b	'3DRT_256.PAl',0	*パレットデータ
spfile:		.dc.b	'3DRT_256.SP',0		*スプライトデータ
sintblfile:.dc.b	'sintbl.dat',0		*sinテーブルデータ

		.stack
		.even
mystack:	.ds.l	256
mysp:

	.end
```

リスト2

```
0000  25 9E 2D 6C 68 30 2D 20  : 41
0008  00 00 00 20 00 00 00 13  : 33
0010  9C 24 19 20 01 0C 33 44  : 7D
0018  52 54 5F 32 35 36 2E 50  : 20
0020  41 4C AD CE 48 00 00 00  : 50
0028  00 63 1B 6B 5C 6B 5D 73  : 80
0030  9E 73 9F FF E9 07 C1 FF  : 5F
0038  FF F8 01 E0 01 FF FF 08  : DF
0040  43 00 3F 07 C1 FF C1 24  : 2E
0048  7D 2D 6C 68 35 2D 1E 0D  : 0B
0050  00 00 00 80 00 00 24 0E  : B2
0058  2E 19 20 01 0B 33 44 52  : 3C
0060  54 5F 32 35 36 2E 53 50  : 21
0068  E5 B3 48 00 00 07 DB 73  : 35
0070  77 B2 48 E2 FF 0E E7 81  : C8
0078  F7 77 8E 6E ED D6 65 DB  : 6D
-------------------------------------
SUM:  86 B1 28 6B 4F 5B 6C F1  9D8F

0080  9B 59 49 5F 83 18 96 78  : 45
0088  1B 9A EF 6D 2B F0 2A 70  : C6
0090  E2 D4 BF FF FF FD 55 55  : 1A
0098  5D E6 66 E6 6C DD DD DD  : 92
00A0  D9 37 7A 6C E2 74 93 BB  : 9A
00A8  A0 0A 22 44 12 2E 53 84  : 27
00B0  38 8C 04 3E E7 82 47 20  : D6
00B8  9D 01 21 1E 01 0D 86 C3  : 34
00C0  01 90 8F 00 46 43 41 E0  : CA
00C8  D4 AE 99 EB 64 B6 B4 EC  : C0
00D0  2A B6 FB 2C BA C3 51 D3  : A8
00D8  F3 EB AF 6A 5F A1 A7 6D  : 0B
00E0  97 86 9F 5E B8 A8 E9 E9  : 4C
00E8  85 7F 7A D0 08 A0 4C FF  : 41
00F0  8A DE A4 B7 A8 2E 54 02  : EF
00F8  50 08 55 D9 91 A8 6A A8  : D1
-------------------------------------
SUM:  2B 45 02 FC B1 8E 85 DA  C4C1

0100  95 13 6B 2B 97 8F 2B D0  : 5F
0108  57 48 82 ED D9 58 5D CA  : 66
0110  EC AC EA 0E 17 93 15 CD  : 1C
0118  16 69 A6 8A 6D 6A CB FA  : 4B
0120  92 A8 DD 41 78 DB 52 BB  : B8
0128  3F 5D EF 8B 96 C5 65 DC  : B2
0130  AF AC 6F BF 92 82 FF 5A  : F6
0138  71 CD 2F E3 50 E2 49 F7  : C2
0140  71 C0 F3 9C DB 3A 7E A5  : F8
0148  1F 22 D1 46 38 3C D3 B3  : 52
0150  07 95 14 FA 6D 1C DB FA  : 08
0158  FE 3E 12 06 0F 5E 9C A1  : FE
0160  F8 3C 32 3C 3D E6 ED FE  : B0
```

```
0168   FF 1D 34 86 1D A6 20 FC  : B5
0170   8B 22 B4 86 1D 5A 62 79  : 39
0178   3B 6F FC B5 5A 63 79 2B  : BC
---------------------------------------
SUM:   31 8D E7 FD 44 21 17 DA   E258

0180   83 01 A1 5F EA 91 F9 F6  : EE
0188   62 FD F1 7D 50 3E BE DB  : F4
0190   6E 4D F5 A5 B0 2D 9D 97  : 66
0198   AE C4 6F FB B4 9F B6 B7  : 9C
01A0   E7 AF 5D 60 56 B6 E1 D9  : 19
01A8   85 BC 92 D3 C7 AF 1C BC  : F4
01B0   F0 3F FC F3 B4 78 7E 95  : 5D
01B8   51 E3 B3 E8 54 5A 41 D9  : 97
01C0   81 FD 12 A8 CB 10 3C EE  : 3D
01C8   20 6A F6 51 72 E6 0F 2B  : 63
01D0   0B 20 35 00 EC E0 EF BF  : DA
01D8   13 BF A2 5B E3 C5 B7 03  : 31
01E0   8A 45 21 28 36 83 32 36  : 39
01E8   A2 3D AD F3 7D 2D BD 42  : 28
01F0   DE 9A 35 BB 90 5D CE 2E  : 51
01F8   EA 97 A3 3F CC 98 E0 D9  : 80
---------------------------------------
SUM:   61 95 19 F3 DE 12 54 7C   4788

0200   C9 66 E8 CB 0E A3 45 F3  : CB
0208   D4 C1 C8 DB 28 9D 3F D7  : 13
0210   78 8C 79 0C 95 AF F6 9C  : 5F
0218   FA F6 4B 17 B6 B2 6F CC  : F5
0220   47 8A F1 59 D7 D9 7A 07  : 4C
0228   AE 28 BC 6F F7 DE FB 3D  : 0E
0230   E0 95 F7 00 DE A9 04 25  : 1C
0238   7D 5D A5 2E E6 4D 28 F5  : FD
0240   E8 54 42 14 21 0F 3F 0B  : 0C
0248   E8 CC 3C 0F DB 95 CA FF  : 38
0250   08 E6 4F FC 42 17 20 F9  : AB
0258   14 19 A4 E5 61 7F 4E E2  : C6
0260   5A B3 CC 7C 3C 3F A0 3D  : AD
0268   35 A2 9A F4 87 44 45 6B  : E0
0270   00 5F 56 1D DF 50 2D 33  : 61
0278   0A 90 30 D6 B3 A2 CC B5  : 76
---------------------------------------
SUM:   E6 B0 1A 26 07 FD DF 05   6B70

0280   9F A4 04 75 B8 10 74 8A  : 82
0288   8E 56 B5 1C CE 98 01 02  : 1E
0290   15 0A FD CF 7F A8 9E 5F  : 0F
0298   CE F9 BC EB 7B BE 90 7A  : B1
02A0   7E 8A 8C 87 1B 1E A7 A7  : A2
02A8   46 7A 34 89 3E 51 E8 79  : 6D
02B0   DC 54 21 67 58 3D 2A 62  : D9
02B8   31 DB 19 17 1C BA DE 7E  : 6E
02C0   6C 5D CC 6C 50 CC C7 9D  : 81
02C8   CD 12 DE 7F 34 5E 07 A1  : 76
02D0   B1 A3 FE 07 99 F9 5E 56  : 9F
02D8   B7 41 A3 6D D8 F2 F0 67  : 29
02E0   DC 73 E0 02 7A 1F C0 8C  : 16
02E8   6C F9 2E 58 82 DD 14 ED  : 4B
02F0   05 58 BC BC CC 46 0D 08  : FC
02F8   A4 1A 28 45 9B F0 B9 98  : 07
---------------------------------------
SUM:   73 61 A9 93 A5 BB F0 79   B01C

0300   8C 0E A5 19 98 F7 77 32  : 90
0308   77 1E 9C 32 8A 62 0C 72  : CD
0310   22 36 99 E3 FD 74 CE 17  : 2A
0318   37 88 71 E4 7D 34 87 80  : CC
0320   AD 5F 98 41 83 99 FD D4  : D2
0328   5C 01 A3 D5 7D 1F 3B C0  : 6C
0330   47 F9 74 2A 83 FA AC 7D  : 7E
0338   33 2F 55 C7 D5 F3 77 F2  : AF
0340   63 FA 37 E4 F1 76 07 AE  : 94
0348   49 AA 0F 52 C3 27 98 3C  : 12
0350   3C CD B1 7A BC C8 97 58  : A7
0358   95 F7 28 79 BF 3C 65 EB  : 78
0360   64 BE 90 83 D0 70 15 DC  : 66
0368   B9 2B E2 58 95 CA BF 02  : 3E
0370   FF A5 11 6A 00 BC 0E 52  : 3B
0378   2F 53 94 11 A5 F8 D7 43  : DE
---------------------------------------
SUM:   A7 BB 85 98 2D 2F 87 DE   F2B3

0380   6E B6 EC 7E 0B 75 F7 D8  : DD
0388   DF BC 57 F5 A9 F0 71 19  : 0A
0390   4C BE 03 1A 97 07 5C C3  : E4
0398   1C 6B AB C0 CE 81 71 0B  : BD
03A0   50 59 0B 92 84 D7 C7 19  : 81
03A8   C4 B6 C5 B3 B4 6C F8 5D  : 67
03B0   E0 F0 CF EA FA 72 7C 7E  : EF
03B8   72 7C 78 9C BC 9E DE 6F  : A9
03C0   73 0D 20 39 F8 31 BF 42  : 03
03C8   3E 8C 57 3E 50 39 1D 02  : 07
03D0   72 31 7C 5E 64 23 5A 11  : 6F
03D8   DD 1E CE D7 B5 CB D2 F5  : E7
03E0   36 3D 5F 59 1C FE A6 14  : FF
03E8   C6 C7 A9 AD 8E AD 24 DC  : 1E
03F0   FB 73 58 E5 5D 8A 55 BD  : A4
03F8   CD 84 7F 9C F9 32 73 03  : 0D
---------------------------------------
SUM:   DF F9 A8 4B 68 FF E8 1C   69C6

0400   18 C5 67 57 84 CB 6A CA  : 1E
0408   0C 7F A0 18 CC 44 6F 51  : 13
0410   92 1D 48 49 B3 6B F8 3C  : 92
0418   CC 22 8E 14 51 B0 66 5E  : 55
```

```
0420   45 19 2B DF 63 EB 8A C7  : 07
0428   0A 85 7B 00 FB D0 F1 C1  : 87
0430   1B 9E C5 1B 83 56 AA 3D  : 59
0438   2F B4 8C 85 B5 29 FB FF  : CC
0440   CE 01 97 42 FF E6 A4 80  : B1
0448   57 82 CC 70 0E F2 00 C0  : D5
0450   BB 18 57 CB 00 DE B7 04  : 8E
0458   F0 64 33 8F B2 EC 91 8E  : D3
0460   10 29 E4 01 73 8B 00 FD  : 19
0468   5D 77 AD 61 F9 85 42 BD  : 5F
0470   EF B3 7B ED D0 80 E4 41  : 7F
0478   33 7D 3B DF 56 0E 9D 17  : E2
---------------------------------------
SUM:   7A 42 08 85 3B A4 06 5D   7444

0480   E1 AA 22 76 45 15 45 F3  : B5
0488   46 B3 C2 31 FB CE 56 0B  : 16
0490   1C DF 41 BB 63 FA 4E 2B  : CD
0498   B1 F2 F3 A4 47 38 A8 EB  : 4C
04A0   7C 39 A8 90 2D 55 E5 5B  : AF
04A8   E3 CA 23 B8 FB 8E 31 1D  : 5F
04B0   D5 A6 40 D7 12 08 8F BD  : F8
04B8   79 1C F8 38 44 25 85 7F  : 32
04C0   C6 18 E7 31 E9 94 B0 B2  : D5
04C8   5C E6 3E FD 1B 20 1D F1  : C6
04D0   D6 67 47 8C 60 F6 D8 33  : 71
04D8   1C C5 00 DF 3A 40 2B 16  : 7B
04E0   35 84 0A 72 21 DA F3 3E  : 61
04E8   31 F7 E0 01 59 5C A0 C6  : 24
04F0   F9 4F DE CC BC C3 DF C2  : 12
04F8   43 3F CB 00 FD 07 9E 44  : 33
---------------------------------------
SUM:   57 26 1A 35 39 0F 9B BE   2BF1

0500   A4 9D 72 18 C9 8C 72 B2  : 44
0508   21 97 40 11 69 08 48 19  : DB
0510   72 7E 89 38 E3 0C AD 4A  : 97
0518   42 27 B6 25 E0 33 73 7C  : 46
0520   A4 BD 91 19 97 09 04 F0  : 9F
0528   3C 70 73 E4 F2 C1 E5 6C  : 07
0530   4D 3C 42 A5 B0 89 63 0E  : 1A
0538   DF 28 7E 14 3F 73 69 BF  : 73
0540   D0 13 FD B8 7D 6E 09 B8  : 44
0548   93 8C F1 3C 9D B7 FE A0  : 3E
0550   9D A0 65 FF EB F1 DE 78  : D3
0558   75 A9 FB 5E 5E A9 A4 6F  : 91
0560   B5 C3 93 CE 6A 5E BD B4  : 12
0568   EF 79 9A 78 35 FB 3F 98  : 81
0570   BC 4F 43 B1 B7 D6 36 1D  : DF
0578   BE 56 9D DD 6B 2E B5 6F  : 4B
---------------------------------------
SUM:   18 33 10 61 91 B5 FF D1   C2DA

0580   AF F1 6C 1D 9C DE 37 EB  : C5
0588   3C EB 73 B3 8C 07 FE 99  : 77
0590   06 32 7C 82 09 A9 14 67  : 63
0598   23 5C 3C F4 87 D7 92 B6  : 55
05A0   4B CF A6 B6 9D 6F FB 3E  : BB
05A8   DF 18 68 8C 86 79 6E 10  : 68
05B0   CF 2C 64 33 C1 DB 04 38  : 6A
05B8   CB BB F0 E0 C6 22 35 87  : FA
05C0   0E A4 61 E1 1F D7 B2 22  : BE
05C8   BD CB 51 87 B9 C9 4C 88  : B6
05D0   4E 9F 0C 44 0D 91 15 FC  : EC
05D8   57 FB 9E AB 88 9E C8 8A  : 13
05E0   F6 DD FC E1 FF FE 00 AE  : 5B
05E8   E0 73 2B 14 38 70 D6 CB  : DB
05F0   BB 97 92 D9 DC 08 A0 E7  : 28
05F8   FF 61 ED D7 63 0B 78 DF  : E9
---------------------------------------
SUM:   D8 89 FB 97 45 9A 46 1D   EE2D

0600   EF 75 36 2D 7A FE B1 CB  : BB
0608   72 29 D8 EF DD 7F 2E DC  : C8
0610   DE 7D 1F 5F 7F 13 B9 E1  : 05
0618   DF 4B 5E B7 BA FF F8 56  : 46
0620   A1 92 B1 43 AE 7D 5A E3  : 8F
0628   6F FB 00 81 53 80 4F F8  : 05
0630   E9 6C 07 A9 70 1C CE 02  : 61
0638   BA 48 C7 60 20 FB 2F FD  : 70
0640   12 B0 09 D9 3A 14 A9 03  : 9E
0648   0E DB 87 E3 22 B1 B2 DC  : B4
0650   E3 49 81 66 D3 33 A2 92  : 4D
0658   D6 4F 65 37 B8 EC CF CD  : 01
0660   A8 FC 54 78 3B 68 F0 75  : 78
0668   28 38 3B F3 80 DE A9 02  : 97
0670   FD 9F C1 13 D4 C2 80 BE  : 44
0678   4F 30 CD E6 5B 58 86 2C  : 97
---------------------------------------
SUM:   C6 CD 9D BC F2 E7 A1 57   E13A

0680   DF 8E 0D FE D0 CB F1 4D  : 51
0688   13 90 CB 06 25 8B 74 55  : ED
0690   F8 F6 41 CF 60 65 D8 19  : B4
0698   B2 6D F9 16 1D 5A 4C B0  : A1
06A0   F4 D6 90 71 E9 28 38 F3  : 07
06A8   BD D6 33 AC 33 EB 59 FA  : E3
06B0   40 47 5B 81 06 4E 1F 47  : 1D
06B8   15 B3 33 A6 BF E3 0E 07  : 58
06C0   03 CD B0 F4 75 6B 86 D7  : B1
06C8   13 27 83 B1 73 99 E0 3C  : 96
06D0   8F 6B F3 1E 4F 62 CD C4  : 4D
06D8   BD F8 73 6D 02 5D 14 6D  : 75
06E0   7E DC FB 5F CF 48 BF 0C  : 96
06E8   94 A1 AF B5 56 FC FD 8F  : 77
```

```
06F0   D7 7A C0 F2 F0 AF D3 46  : BB
06F8   FF 7B AD EA 50 7C 23 15  : 15
---------------------------------------
SUM:   EC F0 13 4D F1 8B 40 E0   B4A6

0700   C0 2B 6B EE C0 32 4F 50  : D5
0708   7A E3 16 FB AD 5C D0 AB  : F2
0710   6C 47 DA 25 2E FF 71 5A  : AA
0718   8A 6A DB C2 6E B0 C1 DC  : 4C
0720   BD DD D8 DF 98 FF B0 0B  : A3
0728   86 2F E3 B4 F6 77 6D 7F  : A5
0730   9B 40 4E 2A 39 FC 3B FF  : C2
0738   F7 B7 FF D6 13 B1 A2 02  : EB
0740   5E 7A 32 84 15 77 04 DD  : FB
0748   55 38 32 EF 83 87 A3 EA  : 45
0750   09 F8 37 FE 49 AA 43 25  : 91
0758   AC CB EF 5E A2 ED BC C8  : D7
0760   7C D2 79 66 9E 32 0B BC  : C4
0768   BE 97 56 51 91 0C 16 C5  : 74
0770   59 88 DE BC D2 B8 95 F7  : 91
0778   C3 7E A1 EA F3 E3 F6 AE  : 46
---------------------------------------
SUM:   C3 A6 16 8F 5A CE 9D 96   9701

0780   30 C7 F5 9F 0F C4 BD B2  : CD
0788   D8 FA BB C6 81 A7 39 D8  : 8C
0790   E5 07 5F 26 31 DC FC 0B  : 85
0798   5E B9 6E 70 8E DF F7 1B  : 74
07A0   FE C5 9B 9E D4 55 9B ED  : AD
07A8   34 EF 72 9D 9D E6 BF 61  : D5
07B0   B9 9C 6F A0 FC 3D DD A7  : 21
07B8   BF B6 E5 D8 F1 78 1A 59  : 0E
07C0   2E 33 90 AF E2 78 C3 31  : EE
07C8   C4 79 05 E6 A7 03 BD E6  : 75
07D0   FF 6B BD CD 49 E2 F6 F8  : 0D
07D8   37 7D D7 D5 FF 27 78 57  : 55
07E0   E4 0C 30 55 7A FD DE EA  : B4
07E8   2D F7 9A F2 53 6D 2E B9  : 57
07F0   F7 46 DF 78 69 5E 09 DE  : 42
07F8   1F EE 4A 1B 3F 7B BC 19  : 01
---------------------------------------
SUM:   44 52 FA BF F3 DD F9 FE   C7FB

0800   B0 AF 74 AE 69 77 5E 0C  : CB
0808   3B 6E 7C F4 6C D7 F2 AF  : FD
0810   78 55 3E C2 E7 AF BF EE  : 10
0818   EF BB FC 07 C3 76 ED FF  : D2
0820   14 0B F7 F8 0F 86 CE D7  : 48
0828   2B B7 EF EF FA 77 5E E1  : 70
0830   FC 1D 15 83 E0 F7 6B D3  : C6
0838   B8 A0 8F 2F 5D FD FF 67  : D6
0840   75 D9 04 2E F6 03 CE CE  : 15
0848   4D DC 3D FF 57 91 B5 1C  : 1E
0850   8B EE AF 3B AB 78 95 20  : 3B
0858   15 FA D6 1D 7F FB 2B D9  : 80
0860   C8 50 F5 FF E0 6F F6 B7  : 08
0868   2D CF D7 B3 10 07 E1 BB  : 39
0870   B5 E6 FF 6C 6D EB 76 B7  : 8B
0878   6D A0 0F BC 8E 14 66 FF  : DF
---------------------------------------
SUM:   BE EE 54 63 27 E0 88 A5   3523

0880   6B 85 DA 6F F7 37 6D 8A  : 5E
0888   DA 7B 81 F9 CD FE D7 ED  : 5E
0890   DD 63 5C 1E C7 09 EB DD  : 52
0898   AC DF F0 7F FB A4 6D EB  : F1
08A0   0F 99 41 17 E8 DD B3 BF  : 37
08A8   F7 D9 63 5F BA 9D FF 58  : 40
08B0   02 07 B7 6D 10 7D C4 1F  : 9D
08B8   FD B0 3F DA FE B1 0F FD  : 81
08C0   8D EF F9 C8 FF 78 83 F0  : 27
08C8   F0 1C 6F EF 7F 9B 15 7D  : 16
08D0   F7 DE 1F 3B 21 03 FF 5B  : AD
08D8   FF FC C6 6B EF 99 FF DA  : 8D
08E0   FC 18 39 FF AD E4 18 47  : 3C
08E8   EF 72 9C 80 FF 9A 4D FE  : 61
08F0   E0 B9 94 83 64 90 75 37  : 50
08F8   4C 19 D7 2F 1C 1A 0E 80  : 2F
---------------------------------------
SUM:   5D AC CE 50 F0 61 9F 10   B5C5

0900   AD D2 B8 03 73 65 28 01  : 3B
0908   FC 75 D7 26 FF 09 38 E9  : 97
0910   FE 12 0B 89 60 03 D8 57  : 36
0918   BE 07 FA 03 CD 91 FB C6  : E1
0920   7F 3B 9D EE 7A E9 20 1F  : E7
0928   90 DF F7 14 0F 34 78 43  : 78
0930   63 E2 5A C5 47 90 5D C4  : 5C
0938   6F F4 07 5D CD 9F EE 5A  : 7B
0940   ED 5C 64 32 E0 10 36 41  : 46
0948   03 74 FD F0 B1 F5 28 DC  : 0E
0950   E1 DF F8 54 0F FF F1 0C  : 17
0958   01 28 47 FC 62 86 C1 AB  : C0
0960   A3 39 AE D9 39 65 43 20  : 64
0968   02 86 AC C2 19 12 5A 1E  : 99
0970   B9 91 68 94 24 31 45 9E  : 7E
0978   8F 97 B7 F2 A2 AE 7F 87  : 25
---------------------------------------
SUM:   05 0E A2 6C 56 2E 87 BE   C0F6

0980   8A 4C EC 80 02 30 C6 66  : A0
0988   9C 7B 2D 10 C9 D3 8D 15  : 92
0990   80 4B 66 1D 1B 43 7B A7  : CE
0998   4C 81 20 16 8A 55 A4 7B  : 01
09A0   1C 00 2E AE 84 92 0C 20  : 3A
```

▶「楽をするためにはどんな努力でもする。楽をするための苦労は厭わない」かあ。ははは，なんか身につまされる言葉だなあ。アマグラマの悲しい性なのですかね。
大森　幹雄(20)神奈川県

```
09A8  43 05 CA 1C 70 38 A9 F7  : 76
09B0  C6 5C 78 3F A4 07 E5 D3  : 3C
09B8  01 01 58 68 44 72 89 C6  : C7
09C0  50 78 F3 B0 C8 AB 0E 2B  : 17
09C8  92 8F 47 6B AB 4C 4E 90  : A8
09D0  97 6D 45 2A F6 78 7F 00  : 60
09D8  5F 7F 28 1F E0 6E 7B B6  : A4
09E0  BF CF 9F 5D C0 2B 61 AF  : 85
09E8  C0 D0 96 BD 6C 07 AE D8  : DC
09F0  D9 FE F6 58 47 21 2D AE  : 68
09F8  E8 17 F7 B2 7F 9D A8 B6  : 22
--------------------------------------
SUM:  30 9C 30 BC 87 AB CF A9  49E6

0A00  39 57 F2 58 F7 C0 FF 19  : A9
0A08  FC 64 A3 3F BD D2 36 C4  : CB
0A10  DD F6 92 DA CB 1B C0 57  : 3C
0A18  F1 37 FE AD F5 C1 67 0E  : FE
0A20  AB 31 FB 83 87 1F B5 59  : 0E
0A28  66 AE 5A 0D EB 96 C2 B7  : 75
0A30  8E 04 E5 2C 49 B7 11 C3  : 77
0A38  38 46 F2 4B 61 2E AE 13  : 0B
0A40  5E 12 5E 1E 0E 0E 05 A8  : B5
0A48  5A 83 69 67 0A EF A3 A7  : F0
0A50  C2 1B 31 DD 2D 42 C2 8B  : A7
0A58  A9 F8 D1 28 5A B5 2E FD  : D4
0A60  03 FF C0 FF FF 5D 14 00  : 31
0A68  F5 24 00 F0 D4 D2 1A 83  : 4C
0A70  01 54 77 3A C0 67 C7 5D  : 51
0A78  6F D4 0C 7A 40 5A 14 0F  : 86
--------------------------------------
SUM:  65 04 5D 52 02 EC 33 EE  20DC

0A80  2B D1 53 81 56 60 9B 57  : 78
0A88  24 05 E0 7F 6C A4 00 20  : B8
0A90  59 73 08 25 E6 78 25 E8  : 64
0A98  15 CC 0B 47 96 05 8A 3F  : 97
0AA0  EC B1 75 A9 01 D8 30 31  : F5
0AA8  C1 E7 A0 33 49 D7 9A 12  : 47
0AB0  9D 86 3E 9E 5F 0E 02 A4  : 12
0AB8  78 19 6A 40 6B C3 23 97  : 23
0AC0  AB CA 75 05 0F E7 AF CB  : 5F
0AC8  C4 AF DB 7F 0F 08 14 21  : 19
0AD0  17 2C 0F 27 A2 FF BC 2D  : 03
0AD8  F1 10 5B EB 58 5C C1 72  : 2E
0AE0  56 D5 84 9F F8 54 2B ED  : B2
0AE8  70 D6 B5 74 54 25 99 4D  : CE
0AF0  54 72 C6 77 A9 CB 94 9A  : A5
0AF8  4F 96 26 99 1F 33 0C B7  : B9
--------------------------------------
SUM:  5F B4 E2 DF 7E C2 DD 32  7908

0B00  E7 61 5B CC 9E B5 EC 01  : AF
0B08  0C 7B 96 62 30 6C FE 18  : 31
0B10  CF E4 A6 64 FD EA 66 6F  : 79
0B18  85 75 80 7F AE B3 A1 EB  : E6
0B20  34 14 0C 71 81 7E A1 F1  : 56
0B28  6D 5E 43 6F 50 1D E5 0F  : DE
0B30  16 84 F1 84 B9 51 4B C8  : 2C
0B38  72 06 B9 03 3F 9B 0F F9  : 16
0B40  DD CF 51 3C AC 42 3F 86  : EC
0B48  17 8F 93 93 4D 1B 96 42  : 0C
0B50  05 7E 34 50 26 02 85 E6  : 9A
0B58  9B F3 F9 B7 55 1B AB DC  : 35
0B60  FB 7A 87 93 EA 8B E1 5C  : 41
0B68  AC 42 C9 E7 01 FE F9 3E  : D4
0B70  F8 49 EF 52 C0 AF 84 1C  : 91
0B78  7C 9E AF 2D CB 1F 87 BF  : 26
--------------------------------------
SUM:  1F A3 0F 47 2C 16 BB 33  AA6B

0B80  D0 AF D4 D3 BF CA F9 7D  : 25
0B88  02 C3 FA 87 DB EF 0B EF  : 0A
0B90  D4 6D 4B 06 A9 83 56 C5  : D9
0B98  AC 61 FC 4C 83 90 C9 61  : 92
0BA0  98 8B 0E 25 8F 55 52 09  : 95
0BA8  80 C4 C1 EF 7E 1C 4B 7D  : 56
0BB0  FA D7 30 28 13 4F 32 16  : D3
0BB8  6C 9F 8E 91 8C E4 38 B6  : 88
0BC0  72 99 5A 7C AF D3 F3 47  : 9D
0BC8  EB 95 8F D3 61 68 FE 9A  : 43
0BD0  0E 18 C8 0F F8 C8 1C FA  : D3
0BD8  4D 8A 32 2F A9 08 DE 8A  : 51
0BE0  B6 15 A9 B3 8A D6 C4 B5  : 00
0BE8  B1 6E 15 0A FE 8C 42 F6  : 00
0BF0  05 FE A4 06 D9 C2 03 A9  : F4
0BF8  20 FD AA AD D0 7F F4 6F  : 26
--------------------------------------
SUM:  14 53 91 76 54 1E 12 0C  D01D

0C00  C9 FB 7D 73 79 02 FD F2  : 1E
0C08  FF FD B5 90 42 87 93 DD  : 7A
0C10  04 FF 78 68 F7 87 2F F6  : 86
0C18  5F FF 7C 23 FF FF CF FB  : C5
0C20  F8 FB 36 EE 3E 97 78 BF  : 23
0C28  32 EB 5A 36 1B 9F A5 EA  : F6
0C30  21 D2 68 A4 D9 34 9E 39  : E3
0C38  7E 79 E7 84 F4 A9 76 CD  : 42
0C40  17 F2 45 4F 75 46 B8 B7  : C7
0C48  2E 9B 57 1F 5F 00 07 1B  : C0
0C50  4E 71 77 A7 45 0E 50 9C  : 1C
0C58  E2 F0 4E 7D 47 A4 01 57  : E0
0C60  4C 42 FA B9 29 AD 8F AB  : 51
0C68  51 C4 2B 43 C5 99 E3 E9  : AD
0C70  09 C0 E3 38 2C 37 74 EB  : A6
0C78  7D 9C 2B FD D2 4B EF 19  : 66
--------------------------------------
SUM:  8C 77 99 9D 23 E2 A4 CC  64EA

0C80  6E 5C 0D 96 58 E9 66 1C  : 30
0C88  08 A5 2D 84 D4 A5 B0 D8  : 5F
0C90  D9 77 E3 A6 37 E1 DD 2D  : FB
0C98  43 61 61 2E A3 51 96 E4  : A1
0CA0  FC 69 F3 0F B8 52 60 87  : 58
0CA8  0B 94 A6 32 D9 8C C4 9E  : 3E
0CB0  69 8E 03 C7 DA E6 4F EB  : BB
0CB8  99 7C 53 34 B7 D7 F0 03  : 1D
0CC0  EE B7 E7 94 A4 F4 5D 2B  : 40
0CC8  02 9E 85 81 5F 46 D9 49  : 6D
0CD0  FE 87 D7 D4 90 75 F4 28  : 51
0CD8  E4 3C FF 89 26 FF AD D6  : 50
0CE0  FF 5C 23 FF B6 13 FC 39  : 7B
0CE8  FF 66 7F 83 CD D7 FE 20  : 29
0CF0  0A 93 5A AE F4 71 1F E3  : 0C
0CF8  E2 E6 E1 7A 1C FB 79 C6  : 79
--------------------------------------
SUM:  57 33 8C 46 74 5F 55 8C  AECB

0D00  95 1F 3A 57 84 07 44 3A  : 4E
0D08  88 47 0D CA 11 C2 F2 C4  : 2F
0D10  E9 99 63 00 B4 6E 64 46  : B1
0D18  15 73 CD FF C8 7E FD 04  : 9B
0D20  35 30 86 B7 90 71 FA FC  : 99
0D28  B0 36 3F CB E7 EC 0A 06  : D3
0D30  F4 58 0E 59 06 FB 83 88  : BF
0D38  87 FF 80 5C 09 A8 F6 F6  : FF
0D40  58 FD BB A8 2C 04 B8 6C  : 0C
0D48  C2 F2 2E 75 B0 6D 3C 09  : B9
0D50  AF 66 A7 ED F5 70 16 DC  : 00
0D58  6F 02 ED 88 3F FE 15 0A  : 42
0D60  85 42 A1 50 AF 8B FF F8  : E9
0D68  AB 07 ED A6 9A 00 24 F6  : F9
0D70  ED 7F B9 BF 5C 4C 9A 46  : 6C
0D78  A9 4B 81 6B 8B F8 13 4C  : C2
--------------------------------------
SUM:  79 99 0F 09 D7 63 03 A3  FFB5

0D80  9F D8 20 13 D4 66 B7 1D  : B8
0D88  D9 5F 80 23 60 2D 6C 68  : 3C
0D90  30 2D 00 01 00 00 00 01  : 5F
0D98  00 00 06 AB 1C 19 20 01  : 07
0DA0  0A 73 69 6E 74 62 6C 2E  : C4
0DA8  64 61 74 41 73 48 00 00  : 35
0DB0  00 00 03 23 06 45 09 64  : DE
0DB8  0C 7C 0F 8C 12 94 15 8F  : 6D
0DC0  18 7D 1B 5D 1E 2B 20 E7  : 5D
0DC8  23 8E 26 1F 28 99 2A FA  : DB
0DD0  2D 41 2F 6B 31 79 33 67  : 4C
0DD8  35 36 36 E5 38 71 39 DA  : 42
0DE0  3B 20 3C 42 3D 3E 3E 14  : A6
0DE8  3E C5 3F 4E 3F B1 3F EC  : AB
0DF0  40 00 3F EC 3F B1 3F 4E  : E8
0DF8  3E C5 3E 14 3D 3E 3C 42  : 4E
--------------------------------------
SUM:  B6 E0 33 9C F6 BB 7B 5A  5530

0E00  3B 20 39 DA 38 71 36 E5  : 32
0E08  35 36 33 67 31 79 2F 6B  : 49
0E10  2D 41 2A FA 28 99 26 1F  : 98
0E18  23 8E 20 E7 1E 2B 1B 5D  : 79
0E20  18 7D 15 8F 12 94 0F 8C  : 7A
0E28  0C 7C 09 64 06 45 03 23  : 66
0E30  00 00 FC DD F9 BB F6 9C  : 1F
0E38  F3 84 F0 74 ED 6C EA 71  : 8F
0E40  E7 83 E4 A3 E1 D5 DF 19  : 9F
0E48  DC 72 D9 E1 D7 67 D5 06  : 21
0E50  D2 BF D0 95 CE 87 CC 99  : B0
0E58  CA CA C9 1B C7 8F C6 26  : BA
0E60  C4 E0 C3 BE C2 C2 C1 EC  : 56
0E68  C1 3B C0 B2 C0 4F C0 14  : 51
0E70  C0 00 C0 14 C0 4F C0 B2  : 15
0E78  C1 3B C1 EC C2 C2 C3 BE  : AE
--------------------------------------
SUM:  3C 76 1A 0A FE 22 E2 D6  C465

0E80  C4 E0 C6 26 C7 8F C9 1B  : CA
0E88  CA CA CC 99 CE 87 D0 95  : B3
0E90  D2 BF D5 06 D7 67 D9 E1  : 64
0E98  DC 72 DF 19 E1 D5 E4 A3  : 83
0EA0  E7 83 EA 71 ED 6C F0 74  : 82
0EA8  F3 84 F6 9C F9 BB FC DD  : 96
0EB0  23 9F 2D 6C 68 35 2D 1A  : 3F
0EB8  03 00 00 9C 0E 00 00 58  : 05
0EC0  21 3A 19 20 01 0A 33 64  : 36
0EC8  72 74 5F 32 35 36 2E 78  : 88
0ED0  EA 2D 48 00 00 02 93 6A  : 5E
0ED8  BD F6 AD 2B FE FF F6 B3  : 31
0EE0  DF 76 D6 CE C6 91 11 1A  : 7B
0EE8  7A 6C 88 F0 B7 93 2C AE  : 82
0EF0  AD 4A E6 63 10 5D EB B5  : 4D
0EF8  40 F0 10 F1 EF 10 F4 81  : A5
--------------------------------------
SUM:  BC 6E 14 82 59 80 75 EE  254E

0F00  13 A6 74 8B 18 2E F5 CA  : BD
0F08  5C 25 CC 64 22 E0 DC 5B  : EA
0F10  87 41 70 F5 C2 DB 4E 06  : 1E
0F18  A4 24 58 3D 78 F7 06 23  : F5
0F20  21 B8 09 EF 78 4F 7B C2  : D5
0F28  E3 0F 7F FD F6 DB AA 17  : 00
0F30  26 E2 5F 96 17 02 39 FD  : 4C
0F38  DE 99 B4 E3 BD 6C 3E D8  : 4D
0F40  72 7E 71 3C 51 70 04 D2  : 34
0F48  BA 15 E6 87 DB F7 62 9B  : 0B
0F50  8A 29 70 81 25 D8 A9 E8  : 32
0F58  45 27 F0 41 B2 8E 91 49  : B7
0F60  4D 54 52 85 E6 5D 07 CA  : 8C
0F68  C2 4E 1E 79 75 75 FA 79  : 04
0F70  12 BF 38 23 20 80 BE 47  : D1
0F78  14 B0 41 7C 14 BF B0 6D  : 71
--------------------------------------
SUM:  D2 66 43 A8 48 56 D0 91  7789

0F80  87 C2 57 54 72 F2 40 52  : EA
0F88  40 1F FC 62 6C 2F 52 B4  : 5E
0F90  9F F3 5E 4C 80 59 3D 36  : 88
0F98  24 EB 1F EF AE 7A A1 F4  : DA
0FA0  D4 D7 B7 AB AA E2 70 8C  : 95
0FA8  2D 25 03 39 CE 6D BC 77  : FC
0FB0  31 B9 4E 9D 52 3D DC D7  : 17
0FB8  56 93 60 83 97 73 62 7B  : B3
0FC0  17 FF 37 2A 0D 23 C1 13  : 7B
0FC8  DA BD 19 62 CB 9E 08 9E  : 21
0FD0  E5 FE 25 0D C8 44 D9 5F  : 59
0FD8  67 9A 53 49 29 B0 AD C8  : EB
0FE0  AF 99 59 F6 EF 6A 1A FA  : 04
0FE8  26 A7 F4 C3 3B FC E9 05  : A9
0FF0  60 C9 24 62 62 53 6F 09  : DC
0FF8  4C A4 52 8C 11 C7 88 D9  : 07
--------------------------------------
SUM:  D0 08 C3 7E D3 28 23 3E  3475

1000  C1 18 14 E9 22 97 58 0F  : F6
1008  13 53 3A 2C 02 E9 A2 D8  : 31
1010  26 77 66 BB 7C A4 EC 9D  : 67
1018  BE 00 F4 C1 5E 9A 7D 1E  : 06
1020  D8 B2 88 D7 A4 D5 12 5A  : CE
1028  57 24 0E 4D 0E D3 04 F5  : B0
1030  C5 94 DD 65 B6 E3 04 B2  : EA
1038  A3 1F 60 9B 58 AF 80 82  : C6
1040  C9 C9 B5 88 2C 96 77 70  : 78
1048  BB 3B 78 E1 64 65 97 83  : 32
1050  FA 74 AE EE EE 83 E1 04  : 60
1058  30 9F 8A EC 86 1F 5E 08  : 50
1060  78 27 4F F5 CF F9 2B F2  : C8
1068  5B E6 AF CD 6F 14 C2 DC  : DE
1070  72 CA 9D FC 33 71 2D EE  : 94
1078  C0 B7 49 19 83 4A 5C 0D  : 0F
--------------------------------------
SUM:  02 10 C4 CF B6 5D C0 ED  EFA9

1080  A6 D5 14 8D 68 82 E8 3C  : 2A
1088  35 FE 5B CD 74 31 E5 29  : 0E
1090  E7 37 F0 5B A2 FF 05 AE  : BD
1098  1F 5D D4 D7 E8 EE F8 25  : 1A
10A0  DD E8 AB 45 67 B3 47 61  : 77
10A8  FF DE CE 14 D3 DE 57 79  : 40
10B0  6B 90 BA 25 19 34 74 76  : 11
10B8  A4 ED E8 ED 1A 8F DB 35  : 1F
10C0  1E 68 7E 22 C3 CA 69 57  : 73
10C8  8C 68 67 96 9B EF 9F 8A  : A4
10D0  3B 2E A6 F1 F0 2A F2 EF  : FB
10D8  27 A8 2C B4 33 36 A3 CB  : 86
10E0  FD 12 B3 73 20 27 2C 17  : BF
10E8  42 9D 23 70 E5 A7 FA 24  : 1C
10F0  AD 58 9A 06 26 87 21 3A  : AD
10F8  15 E4 E4 9A 4F C2 08 40  : D0
--------------------------------------
SUM:  D9 3B 59 D7 CE 24 A3 0D  E742

1100  2D 46 F0 A9 A6 B5 09 34  : A4
1108  21 CD 87 3B 75 5D D5 AE  : 05
1110  0A 0F 16 B2 B7 93 C6 44  : 35
1118  EB DD 02 BF 29 A6 B2 C6  : D0
1120  7C F2 D9 B7 CD 2C DD DA  : AE
1128  5D E0 05 F6 BD F6 BE 00  : A9
1130  E6 E4 5D 6C 26 43 64 36  : 96
1138  43 64 3F B8 7E 3A 85 EF  : CA
1140  93 E4 32 6F 66 1B A9 09  : 4B
1148  7A 8F 9F 2E 2F C1 88 BB  : 09
1150  F6 B7 ED 60 0E 32 1B 21  : 76
1158  B2 1B 21 FE 03 FD C7 A8  : 5B
1160  8F 2C 3F 83 27 1F 17 D5  : AF
1168  20 8D 72 43 FE F2 A3 DE  : D3
1170  3E 60 7E 45 F2 79 4F 22  : 3D
1178  BF 96 5A 76 B5 BD 1B E9  : 9B
--------------------------------------
SUM:  A6 0D 71 A2 9B 3C 11 36  2DA8

1180  05 A3 0C DB 17 CA 9D C4  : D1
1188  D1 83 27 1C 19 18 83 27  : 72
1190  24 1B DA 84 62 74 95 E7  : EF
1198  73 2F 3A B1 7B 5F A3 FE  : 08
11A0  85 F3 EC 6C 3C A0 61 CE  : DB
11A8  CE 30 66 77 72 E3 FE D9  : 07
11B0  4F BB 9F AC 71 F4 49 0E  : 11
11B8  78 3A 0B 10 B3 14 D5 C3  : 2C
11C0  82 10 FA 36 1E 0D 87 1F  : 93
11C8  83 70 B5 1C 76 0F 04 13  : 60
11D0  82 E4 7E D5 C7 E3 F5 D8  : 30
11D8  E1 C3 33 54 16 EC 4A BB  : 32
11E0  15 D5 FA 71 96 39 60 2D  : B1
11E8  D8 20 25 DB 0B B6 C0 00  : 79
11F0  00 00 00 00 00 00 00 00  : 00
11F8  00 00 00 00 00 00 00 00  : 00
--------------------------------------
SUM:  DC A4 C2 92 F1 1A BF 3A  1C94
```

▶10月号LIVE SPECIALの「笑顔を探して」はすごい出来だ。でも早く聞きたいがために人間プリプロセッサをやってしまった。やっぱり「ZPP.X」を入力したほうがよかったかなあ。
美辺　央希(18)東京都

WE WANT YOU!

Oh!Xの掲載記事を理解するうえで重要となるキーワードに「パーソナルコンピューティング」という言葉があります。なにも，難しい概念などではありません。Oh!Xが提唱しているのは，「パーソナルコンピュータをちゃんとパーソナルコンピュータとして使う」，というごく単純なことにすぎないのです。

それぞれの人がそれぞれのスタイルでパーソナルコンピューティングを楽しんでいると思います。それがどんなものであるかを知ることは，本誌の誌面作りにとって非常に重要なことなのです。そして，Oh!Xが発信したメッセージを皆さんが受け取り，それに対する皆さんのメッセージが今後のOh!Xの方向を決めていくことにもなります。

実際，Oh!Xの誌面はスタッフだけが作っているものではありません。これまでのOh!MZ/Xの軌跡をたどると要所要所で読者投稿作品が大きな影響力を及ぼしていることがわかります。読者の力がこれまでのOh!Xを支えていたといっても過言ではないでしょう。

しかし，影響を与えられているのは投稿作品だけではありません。実はそれ以上の影響力を持つのがアンケートハガキによるメッセージです。Oh!Xの全体的な方向性を決めているのは誌面にはあまり現れない多くの人の意見なのです。読者層が変われば記事が変わる，というほど単純なものでもありませんが，記事の方向性に多大の影響を及ぼしています。

投稿作品はそれ自体が強いメッセージでもあります。強いメッセージは歓迎します。また，アンケートハガキの回収にもご協力ください。多くの方の意見が揃ってこそ，よりよいフィードバックが行われます。

私たちはいつでも皆さんからのメッセージを求めています。

〈重点募集項目〉

●オリジナル音楽データ

●カードゲーム

●SX-WINDOWアクセサリ

●Z's-EX用外部コマンド

イラスト投稿の規定

サイズはハガキ大（A6判）以上であれば可。B5判くらいまでは可能ですが，取り扱いの手間や現実的な問題としてハガキ大を一応の標準とします。いずれにせよ，掲載時にはかなり縮小されることを考慮して描いてください。

一応の推奨形式は以下のとおりです。

1）ハガキ大のケント紙で郵送

ハガキでも結構ですが，たまに裏面にも消印が押される場合があります。

2）黒1色（薄ズミ不可）

墨汁は汚れの原因になることがあります。製図用インクがおすすめです。原稿は縮小されますのでスクリーントーンの80，90番台（レトラセットの場合）などや色の濃すぎるものについては再現は保証されません。残念ながら，カラー原稿はごくたまにしか掲載されません。

内容に関して特に規制はありませんが，時期もの（正月，クリスマス，季節もの）などについては，掲載が予想される時期を考慮して早めに送ったほうが有利になることがあります（年賀状は例外）。

それでは，皆さんの力作をお待ちしています。

協力スタッフ募集

Oh!Xでは誌面作りに参加していただく協力スタッフを募集しています。

スタッフとして活動する熱意があり，東京近郊にお住まいの方でソフトバンクまで来社可能な方。特に時間的な束縛はありませんが，ある程度時間的な余裕がある方に限ります。基本的に学生を対象としていますが，十分に時間的余裕と余力があれば社会人も可とします。ただし，18歳未満の学生および浪人生の方については採用予定はありません。

応募要項です。ライター希望の方はOh!X誌面2ページ分相当（2000字程度）の自由論文に自己紹介文を添えて「Oh!Xスタッフ希望」係までお送りください。

また，文章力には自信がないけどプログラムなら……という方でも技術スタッフとして，参加していただく場合があります。こちらを希望の方は自由論文の代わりに，これまでに制作した自作プログラムとその解説などを一緒に応募してください。

書類選考後，採用者の方にはこちらから連絡いたします。

投稿大募集

Oh!Xでは読者の皆さんによる投稿作品を常時募集しています。

未発表の作品であれば，グラフィック，音楽，システムプログラム，ツール，ゲーム，ハードウェアなどジャンルを問いません。数当てゲームからOSまでなんでも受け付けています。機種についても（メーカー，年代など）特に限定はしませんが，雑誌の性格上扱いにくい場合もあります。

誌面に載りきらない大きなアプリケーションなどはディスクメディアを使って配布することが考えられます。その形態のひとつはご存じ付録ディスク，そしてもうひとつは別冊形式によるものです（10月発売予定のZ-MUSICシステムに続き，今後もいくつかのOh!X MOOKシリーズが予定されています）。

また，特に掲載されることを目的とせず，「こんなものを作ってみました」といったプログラムでもかまいません。気軽に作品を送ってみませんか。

投稿募集要項

1）お送りいただくプログラムには，住所，氏名，年齢，職業，連絡先電話番号，機種名，使用言語，動作に必要な周辺機器，マイコン歴などを明記のうえ，封書の宛先の最後には「Oh!X LIVE」，「全機種共通システム」，「投稿ゲームプログラム」など，プログラムの内容を明確にご記入ください。

2）投稿されるプログラムには詳しい内容を記入した原稿を同梱してください。ディスクの中にドキュメントファイルの形式でのみ記述している方がいますが，郵送時の事故などでメディアが破壊されることもありますので，必ず文書を添えるようにしてください。一緒に変数表，メモリマップ，参考文献などがあればなお結構です。また，掲載に際してお送りいただいたプログラムやデータ原稿については，当方で加筆修正をさせていただくことがあります。

3）お送りいただくプログラムは事故防止のため最低2回はセーブしておいてください。基本的に同封されたフロッピーディスク，カセットテープ，クイックディスク，原稿などについてはご返送いたしませんので，あらかじめご了承ください。

4）ハード製作関係の投稿につきましては，最初は内容のわかる原稿のみお送りいただければ結構です。その後，当方で製作物が必要だと判断した場合には改めて連絡いたします。

5）お送りいただいた作品の採用につきましては，掲載号が決定した時点で当方より連絡いたします。特にツール関係，ハード関係などのものにつきましては特集内容などを考慮したうえで採用決定されますので，結果を連絡するまでに時間がかかる場合があります。

6）投稿いただいたプログラムにバグなどが発見された場合は新しいプログラムの入ったメディアと一緒に文書にてご連絡ください。

7）掲載されたプログラムに対しては当社規定の原稿料をお支払いたします。また，投稿されたプログラムの著作権などはすべて制作者に保留されますが，いわゆる「PDSなどとしてネットにアップする」ことなどを希望される場合には必ず事前に編集部までご連絡ください。なお，一般的モラルとして，他誌との二重投稿または，他誌に掲載されたプログラムの移植などについては固くお断りいたします。

その他，不明点については編集室まで問い合わせてください。

宛先

〒108　東京都港区高輪2-19-13　NS高輪ビル

ソフトバンク株式会社

Oh!X編集室「投稿プログラム」係

X68000CARDDRV用カードゲーム

サーティーン・ダウン

Ookubo Akihiro **大久保 明弘**

操作は簡単，プログラムの入力も簡単，もちろんルールも簡単。でも目的を達成するためには，かなりの運が必要となる「サーティーン・ダウン」。４組のシークエンスを完成するまで，気合を入れて遊びましょう。

「サーティーン・ダウン」は，台札をベースに各スートの数上がりシークエンスを作るのが目的のカードゲームです。難易度は結構高いので，あきらめず何度もトライしてください。

入力方法

リストは，X-BASICで記述されていますので，BASICを立ち上げてそのまま打ち込んでもらえば結構です。

このゲームを実行するためには，CARDDRV.X（もしくはCARDDRV2.X）とトランプデータ，CARD.FNC(CARD2.FNCでも可）が必要になりますので，事前に組み込んでおいてください。

なお，なんの変更もなくコンパイルすることができます。

遊技法

使用するカードはジョーカーを除いた52枚です。

まず，カードをシャッフルして上の13枚を裏向きに場に置きます。この13枚が予備札になります。次に予備札の左と右にそれぞれ４枚ずつカードを表向きに置きます。これが場札となります。さらに予備札の上に台札としてカードを１枚表向きにセットします。残りのカード30枚が手札となります（図１）。

最初にいったとおり，目的は台札としてセットした数から始まる，４組の数上がりのシークエンスを作ることです。

プレイヤーは，まず８枚の場札の中に台札と同じカード，もしくは台札に続くカードがあれば移動できます（図2-1）。そして，台札を移動して空いたスペースには，予備札から１枚カードをめくり，置くことができます（図2-2）。

台札として置けるカードは，初めに置かれた場札の数字と同じでなければなりません。

場札だけではすぐに行き詰まるので，行き詰まったら手札を使います。手札が台札として使えるか，シークエンスが使えるのなら，場札にあったカードと同じようにそのカードを台札に重ねることができます（図３）。

使おうとした手札がどこにも置けなかった場合は，そのカードは捨て札となります。捨て札は手札の下にある，捨て札のエリアに置かなくてはなりません。ただし，捨て札のトップカードは使うことができます（図４）。

そして，手札を全部使っても４組の数上がりシークエンスが完成しない場合，捨て札をシャッフルして再び手札として使うことができます。この捨て札を使える回数は，ゲーム開始後に設定するレベルによって０～５回と決まっています。

操作方法

操作にはマウスを使います。プログラムを起動するとレベル選択画面になりますので，カーソルを動かして左クリックしてください。

各レベルは，
EASY……捨て札を５回使える
NORMAL……捨て札を３回使える
HARD……捨て札を使うことはできない
以上のような設定になっています。

カードを移動させる場合には，移動させるカードにカーソルを合わせ左ドラッグします。そして，ドラッグした状態で移動先に持っていき左ボタンを離せば，カードはそこに移動します。

手札や予備札を使う場合は，裏返しになっているカードを左クリックし，ドラッグしながらカードを移動させてください。

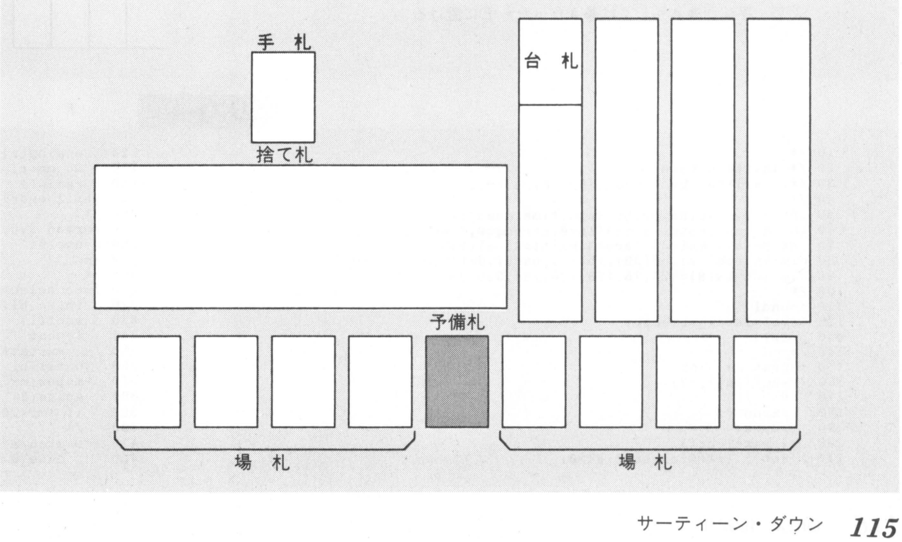
なお，リトライする場合は，画面左上にある“retry”を左クリック，ゲームをやめたい場合には，左上にある“exit”を左クリックしてください。

プログラムについて

いつものとおり，表１に変数表を用意しておいたので，それを参考にしながらプログラムを見てください。

また，参考にした本には「予備札は場札にしかできない」と書いてありました。要するに予備札を開いたときに，台札へ重ね

図１



ることができるカードが出た場合でも，一度場札のエリアに置いてからでないと台札に重ねることができないということなのです。

しかし，これでは面倒なだけですので，ダイレクトに置けるようにしました。もしも，「そんなことではイカン！」というストロングな人のために，変数strongの値を0から1にすることで，予備札は一度場のエリアに移動してからでないと使えなくなります（そんな人はいますかねえ）。

最後に

このカードゲームの名前「サーティーン・ダウン」というのは少し変ですね。目的は数上がりシークエンスを作るのに“ダウン”とは，これいかに？　地価が高騰しても“ダウンタウン”というがごとし，なんちゃって。

まあ，納得がいかない人のために隠し変数downの値を0から1にすると，タイトルどおり(?)数下がりのシークエンスを作ることを目的とする「サーティーン・ダウン」になります。

てなわけで，「サーティーン・ダウン」は，

「元気モリモリ，森昌子一っ」

「花も咲く咲く，桜田淳子一っ」

「桃もすももも，山口百恵一っ」

な出来です。あいかわらずのギャグを飛ばしていますが，出来自体は保証しますよ（ちょっとしつこいかな）。ぜひ入力して遊んでみてください。

＜参考文献＞

トランプゲーム，藤村孝，高橋書店

表1　変数表

i, j, r	汎用
level	どのレベルか？
dead	捨て札を使える回数の制限用
yp	予備札のポインタ
hp	手札のポインタ
sp	捨て札の数
bc	移動する場札のカード番号を格納
kazu	手札の枚数
time	捨て札を使った回数
d_card	台札のカード番号を格納
nothing	手札があるか？
endflg	ゲーム終了フラグ
cd()	カード番号を格納
hd()	手札の内容
yb()	予備札の内容
ba()	場札の内容
da()	各スートの台札の内容
dp()	各スートのシークエンス状況を格納
sute()	捨て札の内容
bx()	場札のX座標
pasa, chin, tero	効果音用
waitl .wait2	ウエイト用

図2－1

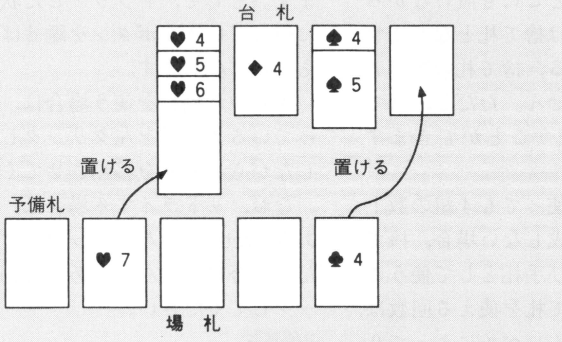


図2－2

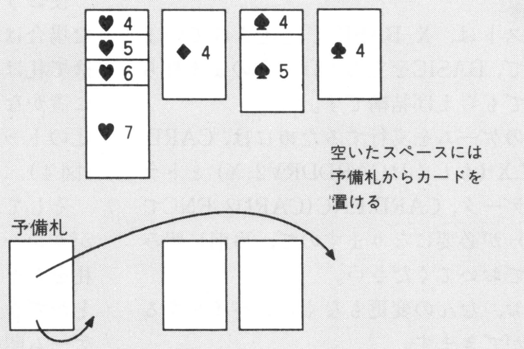


図3

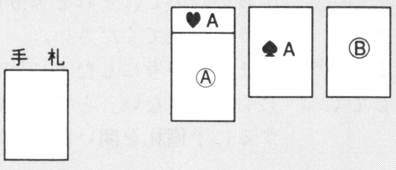
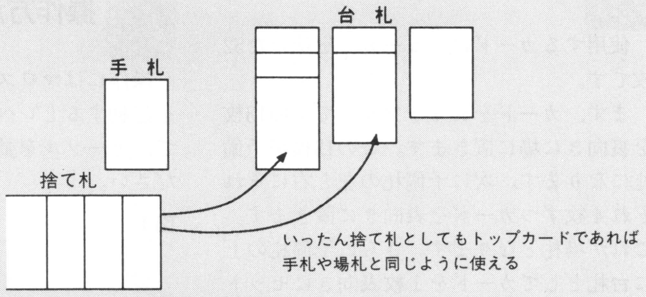


図4



リスト1

```
 10 /*
 20 /* Thirteen Down
 30 /*   Written by Azuron 1992 8.28(Fri.)
 40 /*
 50 int i,j,r,yp,hp,sp,bc,kazu,time,dead,level
 60 int d_card,nothing,endflg=0,strong=0,down=0
 70 int pasa=1,chin=2,tero=3,wait1=4,wait2=5
 80 dim int cd(52),hd(30),yb(13),ba(8),da(4),dp(4),sute(30)
 90 dim int bx(8)={24,76,128,180,284,336,388,440}
100 /*
110 sound()
120 level=level_select()
130 prep()
140 /*
150 repeat /* メイン
160   shuffle(1,52)
170   vinit()
180   layout()
190   repeat
200     r=select()
210   until (r<>0 or endflg<>0)
220   ending(r)
230   Go_on(r)
240   again()
250 until endflg
260 /*
270 screen 2,0,1,1 /* 終了
280 mouse(0)
290 end
300 /*
310 func select() /* カードを選ぶ
320   int n,bl,br,mx,my,r=0
330   msoff()
340   repeat
350     msstat(n,n,bl,br)
360   until bl
370   mspos(mx,my)
380   apage(3):n=point(mx,my)
390   if (my<20 and n<>0) then n=n+15
400   /*
410   switch n
420     case 0
```

```
430     case 10:case 11:case 12:case 13
440       break
450     case 9
460       yobi_open()
470       break
480     case 14
490       if nothing then r=seiri() else hand_open()
500       break
510     case 15
520       sute_use()
530       break
540     case 16
550       if retry() then r=3
560       break
570     case 17
580       if quit() then endflg=1
590       break
600     default
610       ba_move(n-1)
620       break
630   endswitch
640   /*
650   if endcheck() then r=2
660   return(r)
670 endfunc
680 /*
690 func hand_open() /* 手札を開く
700   dim int xx(5)={124,124,124,124,130}
710   dim str m(5)={"NEXT","NEXT","NEXT","NEXT","END"}
720   /*
730   apage(0):c_put(124,110,hd(hp))
740   number(140,210,kazu-hp-1)
750   if hp>=kazu-1 then {
760     if level=2 then xx(2)=xx(4):m(2)=m(4)
770     if level=3 then xx(0)=xx(4):m(0)=m(4)
780     apage(1):fill(124,110,171,205,0)
790     flbx(122,144,172,170,14,15)
800     symbol(xx(time-1),146,m(time-1),1,1,2,15,0)
810     nothing=1
820   }
830   r=card_move(0,148,158)
840   switch r
850     case -1
860       erase(0):number(140,210,kazu-hp)
870       if nothing then {
880         apage(1):fill(122,144,172,170,0)
890         c_put(124,110,0):nothing=0
900       }
910       break
920     default
930       if sp=0 and (kazu-1)=hp then {
940         apage(1):fill(123,145,171,169,14)
950         symbol(xx(2),146,m(2),1,1,2,15,0)
960       }
970       hp=hp+1
980       break
990   endswitch
1000 endfunc
1010 /*
1020 func yobi_open() /* 予備札を開く
1030   int p,f=1,x=248,y=498
1040   for p=0 to 7
1050     if ba(p)=0 then f=0:break
1060   next
1070   /*
1080   while f=0 and yp<>13
1090     apage(0):c_put(232,402,yb(yp))
1100     number(x,y,13-yp-1)
1110     if yp=12 then apage(1):fill(232,402,279,497,0)
1120     p=card_move(2,254,450)
1130     if p=-1 then erase(0):number(x,y,13-yp) else yp=yp+1
1140     break
1150   endwhile
1160 endfunc
1170 /*
1180 func ba_move(p) /* 場札移動
1190   int x,r
1200   while ba(p)
1210     x=bx(p)
1220     apage(1):fill(x,402,x+47,497,0)
1230     apage(0):c_put(x,402,ba(p))
1240     bc=ba(p):ba(p)=0
1250     r=card_move(1,x+24,450)
1260     while r
1270       ba_put(0,p,bc)
1280       break
1290     endwhile
1300     break
1310   endwhile
1320 endfunc
1330 /*
1340 func sute_use() /* 捨て札のトップカードを使う
1350   int x
1360   if sp=0 then return(-1)
1370   x=(sp-1)*8+6
1380   apage(1):fill(x,272,x+47,367,0)
1390   if sp<>1 then sute_put(0,x-8,sute(sp-2))
1400   apage(0):c_put(x,272,sute(sp-1))
1410   r=card_move(3,x+24,320)
1420   switch r
1430     case -1:erase(0):sute_put(0,x,sute(sp-1)):break
1440     default:sp=sp-1
1450   endswitch
1460   return(0)
1470 endfunc
1480 /*
1490 func card_move(sw,zx,zy) /* カード移動
1500   int n,bl,br,mx,my,hx,hy,r
1510   msarea(24,50,485,463):setmspos(zx,zy)
1520   apage(3)
1530   repeat
1540     mspos(mx,my)
1550     if mx<zx then hx=0 else hx=511
1560     if my<zy then hy=0 else hy=511
1570     home(0,hx+zx-mx,hy+zy-my)
1580     msstat(n,n,bl,br)
1590   until bl=0
1600   /*
1610   r=destination(sw,point(mx,my))
1620   msarea(0,0,511,511)
1630   return(r)
1640 endfunc
1650 /*
1660 /* 移動先に応じて、処理を振り分ける
1670 func destination(sw,p)
1680  int r=-1
1690  repeat
1700   if sw=0 then {
1710     if p>9 and p<14 then r=da_card(p-10,hd(hp)):break
1720     if p=15 then suteru():r=0
1730     break
1740   }
1750   if sw=1 then {
1760     if p>0 and p<9 then r=ba_put(1,p-1,bc):break
1770     if p>9 and p<14 then r=da_card(p-10,bc)
1780     break
1790   }
1800   if sw=2 then {
1810     if p>0 and p<9 then {
1820       if ba(p-1)=0 then ba_put(1,p-1,yb(yp)):r=0
1830       break
1840     }
1850     if p>9 and p<14 and strong=0 then r=da_card(p-10,yb(yp
)
1860     break
1870   }
1880   if sw=3 then {
1890     if p>9 and p<14 then r=da_card(p-10,sute(sp-1))
1900     break
1910   }
1920   break
1930 until 1=0
1940   msarea(0,0,511,511)
1950   return(r)
1960 endfunc
1970 /*
1980 func da_card(p,c) /* 台札にできるのなら、新しい台札を表示
1990   int r=0
2000   if d_check(p,c) then r=-1 else da_put(p,c)
2010   return(r)
2020 endfunc
2030 /*
2040 func suteru() /* 手札を捨て札に
2050   erase(0)
2060   sute_put(1,sp*8+6,hd(hp))
2070   sute(sp)=hd(hp)
2080   sp=sp+1
2090 endfunc
2100 /*
2110 func seiri() /* 捨て札を整理して手札に
2120   int r=-1
2130   while (time<>dead) and sp<>0
2140     apage(1)
2150     fill(122,144,172,170,0):fill(6,272,285,367,0)
2160     for i=0 to sp-1:cd(i)=sute(i):next
2170     shuffle(0,sp)
2180     for i=0 to sp-1:hd(i)=cd(i):next
2190     kazu=sp:hp=0:sp=0
2200     time=time+1
2210     nothing=0
2220     c_put(124,110,0):SE(pasa)
2230     number(140,210,kazu-hp)
2240     r=0
2250     break
2260   endwhile
2270   return(r)
2280 endfunc
2290 /*
2300 func number(x,y,n) /* 残数表示
2310   str s[8]
2320   s=itoa(n)
2330   apage(1):fill(x,y,x+15,y+13,0)
2340   symbol(x+(2-len(s))*4,y,s,1,1,1,15,0)
2350 endfunc
2360 /*
2370 func layout() /* レイアウト
2380   apage(1):c_put(232,402,0):SE(pasa)
2390   number(248,498,13-yp)
2400   for i=0 to 7
2410     c_put(bx(i),402,ba(i)):SE(pasa)
2420   next
2430   c_put(292,34,da(0)):SE(pasa)
2440   c_put(124,110,0):SE(pasa)
2450   number(140,210,30-hp)
2460   setmspos(256,256)
2470 endfunc
2480 /*
2490 func d_check(p,c2) /* 台札にできるかチェック
2500   int d,c1,f=1
2510   dim int s(2)={1,-1},A(4)={13,1,1,13}
2520   d=down
2530   while 1
2540     c1=da(p)
2550     if (d_card=rank(c2)) and da(p)=0 then f=0:break
2560     if suit(c1)<>suit(c2) then break
2570     c1=rank(c1):c2=rank(c2)
2580     if (c2-c1=s(d)) or (c1=A(d*2) and c2=A(d*2+1)) then f=
0
2590     break
2600   endwhile
2610   return(f)
2620 endfunc
2630 /*
2640 func da_put(p,c) /* 台札表示
2650   int x,y
2660   erase(0)
2670   apage(1)
2680   x=p*56+292:y=dp(p)*22+34
2690   c_put(x,y,c):line(x+1,y,x+46,y,1):SE(chin)
2700   da(p)=c
2710   dp(p)=dp(p)+1
2720 endfunc
```

```
2730 /*
2740 func ba_put(sw,p,c) /* 場札表示
2750   int f=0
2760   if ba(p) then f=1
2770   while f=0
2780     erase(0)
2790     apage(1):c_put(bx(p),402,c)
2800     if sw then SE(pasa)
2810     ba(p)=c
2820     break
2830   endwhile
2840   return(f)
2850 endfunc
2860 /*
2870 func sute_put(sw,x,c) /* 捨て札表示
2880   apage(1)
2890   c_put(x,272,c):line(x,272,x,367,1)
2900   if sw then SE(pasa)
2910 endfunc
2920 /*
2930 func Go_on(r) /* コンティニュー?
2940   if r=3 or endflg<>0 then return()
2950   if yesno(200,"continue?")=0 then endflg=1
2960 endfunc
2970 /*
2980 func retry() /* 再挑戦?
2990   apage(0):symbol(4,2,"retry",1,1,1,5,0)
3000   r=yesno(224,"Sure?")
3010   wipe()
3020   return(r)
3030 endfunc
3040 /*
3050 func quit() /* やめますか?
3060   apage(0):symbol(473,2,"exit",1,1,1,5,0)
3070   r=yesno(224,"Sure?")
3080   wipe()
3090   return(r)
3100 endfunc
3110 /*
3120 func yesno(x,m;str) /* はい?いいえ?
3130   int n,bl,br,mx,my,r
3140   palet(4,63488)
3150   apage(0)
3160   msarea(170,210,339,305)
3170   box(172,212,337,303,3)
3180   flbx(170,210,339,305,2,3)
3190   box(199,267,241,295,9):flbx(200,268,240,294,8,9)
3200   box(265,267,307,295,4):flbx(266,268,306,294,8,4)
3210   symbol(x,226,m,1,1,2,15,0)
3220   symbol(202,270,"YES   NO",1,1,2,1,0)
3230   msoff()
3240   setmspos(218,280)
3250   /*
3260   repeat
3270     msstat(n,n,bl,br)
3280   until bl
3290   /*
3300   mspos(mx,my)
3310   n=(mx<242)
3320   m_play(tero)
3330   for i=0 to 5
3340     if n then palet(9,0) else palet(4,0)
3350     SE(wait1)
3360     if n then palet(9,63488) else palet(4,63488)
3370     SE(wait1)
3380   next
3390   /*
3400   SE(wait2)
3410   msarea(0,0,511,511)
3420   fill(170,210,339,305,0)
3430   palet(4,1024)
3440   return(n)
3450 endfunc
3460 /*
3470 func endcheck() /* クリアチェック
3480   return((dp(0)+dp(1)+dp(2)+dp(3)=52))
3490 endfunc
3500 /*
3510 func flbx(x0,y0,x1,y1,c0,c1) /* fill&box
3520   fill(x0,y0,x1,y1,c0):box(x0,y0,x1,y1,c1)
3530 endfunc
3540 /*
3550 func erase(p) /* Page0をwipe
3560   apage(p):wipe():home(0,0,0)
3570 endfunc
3580 /*
3590 func vinit() /* 変数初期化
3600   for i=0 to 12:yb(i)=cd(i):next
3610   for i=13 to 20:ba(i-13)=cd(i):next
3620   for i=22 to 51:hd(i-22)=cd(i):next
3630   da(0)=cd(21):da(1)=0:da(2)=0:da(3)=0
3640   dp(0)=1:dp(1)=0:dp(2)=0:dp(3)=0
3650   d_card=rank(cd(21))
3660   yp=0:hp=0:sp=0:kazu=30:time=1:nothing=0
3670 endfunc
3680 /*
3690 func sound() /* 効果音設定
3700   m_init()
3710   for i=1 to 5
3720     m_alloc(i,500):m_assign(i,i)
3730   next
3740   m_tempo(200)
3750   m_trk(1,"q6@59v15o4c8")
3760   m_trk(2,"q6v14@5o5c8")
3770   m_trk(3,"q6v13@15o5l8cev13cev10cev7ce")
3780   m_trk(4,"q1r16")
3790   m_trk(5,"q5r1")
3800 endfunc
3810 /*
3820 func SE(t) /* Sound Effect
3830   m_play(t)
3840   repeat:until m_stat(t)=0
3850 endfunc
3860 /*
3870 func msoff() /* マウスのボタンが離されるまで待つ
3880   int n,bl,br
3890   repeat:msstat(n,n,bl,br):until bl+br=0
3900 endfunc
3910 /*
3920 func rank(c) /* 数を調べる
3930   return((c-1) mod 13+1)
3940 endfunc
3950 /*
3960 func suit(c) /* スートを調べる
3970   return((c-1)¥13)
3980 endfunc
3990 /*
4000 func shuffle(sw,max) /* シャッフル
4010   int i,a,b,c
4020   if sw then for i=0 to 51:cd(i)=i+1:next
4030   for i=1 to 199
4040     a=rand() mod max:b=rand() mod max
4050     c=cd(a):cd(a)=cd(b):cd(b)=c
4060   next
4070 endfunc
4080 /*
4090 func ending(r) /* エンディング
4100   int n,bl,br
4110   if r<>2 then return(r)
4120   mouse(2)
4130   apage(0)
4140   flbx(150,228,357,291,8,9)
4150   symbol(160,234,"congratulations!",1,1,2,15,0)
4160   symbol(192,268,"Hit mouse button",1,1,1,3,0)
4170   msoff()
4180   repeat:msstat(n,n,bl,br):until bl+br
4190   fill(150,228,360,299,0)
4200   mouse(1)
4210 endfunc
4220 /*
4230 func again() /* 再度プレイする場合の処理
4240   apage(1)
4250   fill(122,110,173,223,0)
4260   fill(6,272,285,367,0)
4270   fill(24,402,487,511,0)
4280   fill(292,32,507,397,0)
4290 endfunc
4300 /*
4310 func level_select() /* レベル選択
4320   int n,bl,br,mx,my,c,w=65534
4330   screen 1,1,1,1
4340   console ,,0:locate ,,0
4350   palet(13,0):palet(14,w):palet(15,0)
4360   apage(1)
4370   fill(200,220,311,247,1):fill(200,248,311,275,2)
4380   fill(200,276,311,303,3)
4390   apage(0)
4400   flbx(200,212,311,308,8,9)
4410   symbol(232,220,"EASY",1,1,2,13,0)
4420   symbol(220,248,"NORMAL",1,1,2,14,0)
4430   symbol(234,276,"HARD",1,1,2,15,0)
4440   mouse(4):mouse(1)
4450   msarea(200,220,311,303):setmspos(254,262)
4460   apage(1)
4470   /*
4480   repeat
4490     mspos(mx,my)
4500     c=point(mx,my)
4510     switch c
4520       case 1:palet(13,w):palet(14,0):palet(15,0):break
4530       case 2:palet(13,0):palet(14,w):palet(15,0):break
4540       case 3:palet(13,0):palet(14,0):palet(15,w)
4550     endswitch
4560     msstat(n,n,bl,br)
4570   until bl
4580   /*
4590   m_play(tero)
4600   repeat
4610     palet(c+12,0):SE(wait1)
4620     palet(c+12,w):SE(wait1)
4630   until m_stat()=0
4640   mouse(2)
4650   SE(wait2)
4660   dead=(3-c)*2+1
4670   return(c)
4680 endfunc
4690 /*
4700 func prep() /* 準備
4710   srand(val(mid$(time$,4,2)+right$(time$,2)))
4720   vpage(0)
4730   palet(0,12288):palet(1,0)
4740   palet(13,65472):palet(14,33824):palet(15,65534)
4750   erase(0):erase(1)
4760   symbol(180,2,"Thirteen Down",1,1,2,12,0)
4770   symbol(178,0,"Thirteen Down",1,1,2,13,0)
4780   symbol(132,86,"手札",1,1,1,15,0)
4790   symbol(126,248,"捨て札",1,1,1,15,0)
4800   symbol(231,382,"予備札",1,1,1,15,0)
4810   apage(2)
4820   for i=0 to 3
4830     box(i*56+290,30,i*56+340,397,11,&HAAAA)
4840   next
4850   box(4,268,287,370,13,&HAAAA)
4860   flbx(0,0,47,19,2,3):flbx(464,0,511,19,2,3)
4870   symbol(4,2,"retry",1,1,1,15,0)
4880   symbol(473,2,"exit",1,1,1,15,0)
4890   /*
4900   apage(3)
4910   for i=0 to 7
4920     fill(bx(i),402,bx(i)+47,497,i+1)
4930   next
4940   fill(232,402,279,497,9)
4950   for i=0 to 3
4960     fill(i*56+290,30,i*56+340,397,i+10)
4970   next
4980   fill(122,108,173,207,14)
4990   fill(2,270,287,367,15)
5000   fill(0,0,47,19,1):fill(464,0,511,19,2)
5010   vpage(7)
5020   mouse(1)
5030   msarea(0,0,511,511)
5040 endfunc
```

よいこのSX-WINDOW講座――――――――(第11回)

タスク間通信をためす

マルチタスクの醍醐味を感じさせてくれるのがこのタスク間通信です。SX-WINDOWでは複数のプログラム間で通信を行うためのシステムコールが用意されています。耳新しい概念ですのでしっかりと把握しておいてください。

Nakamori Akira 中森 章

はじめに

シャープからSX-WINDOWの開発支援キットが発売されることになりました。噂によるとXCよりも分厚いマニュアルが付属するということなので，11回に及ぶこの連載の内容なんてあっという間に吹き飛んでしまうかもしれませんね。そもそもこの連載は，本誌の付録ディスクに収録されたSX-WINDOWのドキュメント（最初はまったく理解できなかった）に関し，私が理解したところを少しずつ発表するという目的で開始されました。本家本元の「わかりやすい」マニュアルが出るとなると，この連載の寿命もそろそろかなと思いつつ原稿を書いています。

さて，今回のテーマはタスク間通信です。これまでこの連載で扱ってきたテーマは途方もないもの(結構ページ数を食っていた)が多かったので今回はちょっと息抜きです。タスク間通信は，概念自体はいかめしいのですが，それを実現するプログラムは単純です。気楽に読んでみてください。

タスク間通信とは

SX-WINDOWはマルチタスクのウィンドウシステムです。そこでは，個々のウィンドウが異なるタスクとして並行に動作しています。それぞれのタスクは勝手気ままに動作しているのですが，あるタスクがほかのタスクの動作を制御したい場合もあります。

たとえば，SX-WINDOW上で動作するアドベンチャーゲームを作ることを考えます。このとき，コマンド入力(選択)，メッセージ出力，画像出力を別々のウィンドウに分けて行いたいと考えるのは自然なことです。つまり，コマンド入力ウィンドウの状態に応じてメッセージウィンドウや画像ウィンドウの表示が切り替わるようにするわけです。

これを実現する方法としては，ひとつのプログラムで必要なすべてのウィンドウを制御することが考えられます。しかし，この場合，すべてのウィンドウに対して発生するであろう，移動やアップデートなどのイベント処理をひとつのプログラムの中に書いておかなければなりません。これは，扱うウィンドウがひとつだけの場合に比べて，非常に複雑なプログラムになってしまいます。

そこで登場するのがタスク間通信です。これは，タスク同士が連絡を取り合うことによって，自分の希望する動作を相手に要求できるという仕組みです。タスク間通信を利用すればタスク（ウィンドウ）間の連係動作をすっきりとしたかたちで実現することができます。

タスク間通信を実現する方法はいろいろ考えられますが，SX-WINDOWでは新たな（タスクマンの）イベントの発生として実現されています。つまり，あるタスクが別のタスクと通信をしたいと思うときには，タスクマンのイベント（イベントコードが12または13）を強制的に発生させて，相手のタスクがそのイベントを受け取るのを待つという手法を用いています。このとき，要求する動作はタスクマンのイベントの種類(イベントレコードのwhat2)で指定します。また，イベントに関連した付加情報は引数（イベントレコードのwhomとwhom2)を用いて指定します。そして，通信という概念をはっきりさせるために，これらのイベントレコードの情報はメッセージと呼ばれています。

タスク間通信は，メッセージを送られる相手から見ると通常のシステムイベントが発生したのと同じに見えます。メッセージを送られる側は，タスク間通信をシステムイベントのひとつとして処理しておけばよいのです。つまり，通常の場合と同じく，イベントの種類と引数に応じて，しかるべき処理を行ったあとは，次のイベントを待つループに戻ればよいのです。

タスク間通信を行う関数

それでは，SX-WINDOWにおいてタスク間通信に関係する関数の説明をしましょう。表1にタスク間通信を行うために必要になる主な関数を示します。これらは，すべてタスクマンの制御下にある関数です。

タスク間通信に関する関数は，タスクIDを調べる関数，メッセージを送るための関数，メッセージに対する返事を返す関数の3種類に大別できます。

このうち，メッセージを送る関数は，相手を指定する場合と指定しない場合，返事を期待する場合としない場合によっても分類することができます。これらの場合分けを図1に示します。メッセージを送る関数は目的に応じてこれらのなかから最適なものを使用してください。

それぞれの関数の具体的な使用方法を以下に説明しましょう。

●タスクIDを知る

タスクIDとはタスクごとに割り当てられている固有の番号です。自分のタスクIDは，

TSGetID

関数によって知ることができます。あるいは，(TSGetTdb関数によって)自分のタスク管理テーブルの情報を得ているのであれば，タスク管理テーブル構造体のtskidというメンバで自分のタスクIDを知ることもできます。

さて，(相手を指定する)タスク間通信において，相手のタスクはタスクIDによって指定します。したがって，タスク間通信を行う場合は，最初に相手のタスクIDを知らなくてはなりません。

一般的には，タスクIDはプログラムの名前で検索します。そのための関数が，

TSFindTskn

です。この関数はプログラムの名前とタスクIDを指定して，指定したタスクIDよりも大きなタスクIDを持ち，かつ，名前（ワイルドカードも使用可）が一致するタスクのタスクIDを返します。もし，該当するタスクがなければ負の値が返ります。

SX-WINDOWでは同じ名前のプログラムを同時にいくつも起動することができます。TSFindTskn関数で基準となるタスクIDを指定する意味は，名前が一致するタスクのうち，探し出すタスクに制限をつけるためといえます。名前が一致するタスクが複数ある場合は，基準のタスクIDよりも大きいタスクIDを持つタスクのうち，最小のタスクIDを持つタスクが検索されることになります。

これを逆に利用すれば，該当する名前を持つすべてのタスクを探し出すことが可能になります。つまり，最初は基準となるタスクIDを０（０はシステムのタスクIDなので，通常のタスクIDは１以上）にしてTSFindTsknを呼び出します。そして，新たにタスクが見つかるたびにそのタスクIDを基準にして，一致するタスクがなくなるまで，TSFindTsknを繰り返し呼び出していきます。これにより，名前が一致するすべてのタスクのタスクＩＤを得ることができます。

また，タスクマンには自分と同じ名前を持つタスクのタスクＩＤを返す

TSFindOwn

という関数も用意されています。この関数は同じタスクを２つ以上起動したくない場合に，すでに自分（と同じ名前のタスク）が起動されていないかを調べるために使用します。

●タスクマンのイベントを作る

タスク間通信において，メッセージはタスクマンのイベントとして転送されます。タスク間通信でメッセージを送る関数は，その多くが，メッセージとなるべきタスクマンのイベントを引数とします。このため，タスク間通信の送り手は，まずメッセージ用のイベントを作ることが必要になります。

ところで，タスクマンのイベント（tsevent）は次の構造体によって定義されています。

```
typedef struct tsevent {
    short what;/* イベントの種類 */
    long  whom;/* 引数１ */
    long  when;/* システム時間 */
    long  whom2；/* 引数２ */
    short what2；
             /* タスクマンのイベント */
    short tskid;
             /* 送り手のタスクID */
} tsevent;
```

したがって，タスクマンのイベントを作るためにはこの構造体の各メンバの値を決めてやればいいことになります。

whatは発生するシステムイベントの種類です。whatには，実際にイベントを登録する関数（あとで説明するTSSendMes関数など）によって，E_SYSTEM1（イベントコード12），あるいは，E_SYSTEM2(イベントコード13）が自動的に設定されます。ユーザーが値を設定する必要はありません。

what2はタスクマンのイベントの種類です。イベント（メッセージ）を受け取ったタスクは，この値によってシステムイベントの処理を切り分けます。what2の値は－32768～127までがSX-WINDOWのシステムによって予約されていますから，タスク間通信ではユーザーは128以上の値を設定することになります。

whomとwhom2はイベントの内容を補足するための引数です。what2の値だけでは行うべき処理の情報が不足する場合に付加情報を設定します。それぞれがlong int型なので，単純に考えれば８バイトの情報しか付加することができません。しかし，引数としてハンドルやポインタを渡すことにすれば，どのような量の情報を付加することもできます。

whomやwhom2の値を解釈するのは，SX-WINDOWのシステムではなく，システムイベントを受け取ったタスクです。したがって，送り手と受け手の間で，whomやwhom2をどのように解釈するのかという約束事項ができていれば，そこにはなにを設定してもよいのです。

whenはイベントが発生したシステム時間を設定します。システム時間はイベントマンのEMSysTime関数によって知ることができますが，受け手がシステム時間の情報を必要としないのなら，特に設定しなくてもかまいません。

tskidは送り手のタスクIDを設定します。

表１　タスク間通信に関係する関数

関数	説明
TSMakeEvent(long mes1, long mes2, int what2, int Hmode1, int Hmode2, tsevent *eventrecPtr)	mes1（whom）と mes2（whom2）を指定して，タスクマンのイベントを作成する。
TSCommunicate (int listener, tsevent *eventrecPtr, int mode)	受け側のタスクＩＤを指定してタスクマンのイベントを転送する。イベントコードは13。 他のタスクと通信中のタスクに割り込むことができる。
TSCommunicateS(int listener, tsevent *eventrecPtr, int mode)	受け側のタスクＩＤを指定してタスクマンのイベントを転送する。イベントコードは13。 他のタスクと通信中のタスクに割り込むことができる。システム専用。
TSSendMes(int listener, tsevent *eventrecPtr)	受け側のタスクＩＤを指定してタスクマンのイベントを転送する。イベントコードは13。 通常動作中のタスクにのみ通信できる。
TSAnswer(tsevent *eventrecPtr)	タスク間通信に対する返事をする。
TSPostEventTsk(long mes1, long mes2, int what2, int Hmode1, int Hmode2)	mes1（whom）と mes2（whom2）を指定してタスクマンのイベントを登録する。 イベントコードは12。すべてのタスクにイベントを送る。
TSPostEventTsk2(long mes1, long mes2, int what2, int Hmode1, int Hmode2, int tskid)	受け側のタスクＩＤとmes1（whom）と mes2（whom2）を指定してタスクマンのイベントを登録する。 イベントコードは12。
TSFindOwn(void)	自分と同じ名前のタスクが存在するかを調べて，そのタスクＩＤを返す。
TSFindTskn(char *namePtr, int tskid)	namePtr の名前（ワイルドカード対応）で tskid より大きいＩＤのタスクが存在するか調べて，そのタスクＩＤを返す。
TSGetID(void)	現在動作中のタスクＩＤを返す。

図１　メッセージを送る関数の分類

	相手を指定する	相手を指定しない
返事を期待する	TSCommunicate TSCommunicateS TSSendMes	
返事を期待しない	TSPostEventTsk2 TSCommunicate TSCommunicateS TSSendMes	TSPostEventTsk

すでに説明したように，タスクIDはTSGetID関数などによって知ることができます。しかし，tskidも，whatと同様にイベントを登録する関数によって設定されますから，ユーザーがわざわざ設定する必要はありません。

以上のように，タスクマンのイベントを表す構造体の各メンバに直接値を設定することで，タスクマンのイベントを作り出すことができます。しかし，タスクマンには

TSMakeEvent

という，タスクマンのイベントを生成するための専用関数も用意されています。この関数を利用すれば，タスクマンのイベントを作り出す処理を，より短い行数でプログラムすることができます。

TSMakeEvent関数は6個の引数を持ちます。第1引数，第2引数，第3引数が，それぞれ，whomとwhom2，what2に対応しています。これらの引数で与えられる情報に加えて，システム時間と送り手のタスクIDを求めて，TSMakeEvent関数は第6引数で示されるイベントレコード（へのポインタで指し示されるもの）の中に，タスクマンのイベントを作り出します。このとき，システムイベントの種類（what）にはなにも設定されません。

TSMakeEvent関数の第4引数と第5引数は，それぞれ，第1引数と第2引数がハンドルであるかどうかを示します。厳密にいうと，第1引数や第2引数がハンドルで与えられているとき，その内容を別のハンドルにコピーして使用するか否かを指定します。この値が0以外であると，第1引数や第2引数で与えられる値がハンドルであると仮定し，タスクマンは別の領域を用意して，そこにwhomやwhom2の内容をコピーして使用します。そして，作成されるイベントのwhomやwhom2には新しい領域のハンドルが設定し直されます。

ところで，ここで新たに設定し直されるハンドルは誰が廃棄するのでしょう。おそらくは誰も廃棄しません（少なくともドキュメントに記述がない）。結局はTSMakeEventを実行したプログラム自身がハンドルを廃棄しなければなりません。したがって，ハンドルをコピーして使う場合は，元のハンドルとシステムが勝手に作り出したハンドルの2つを廃棄することが必要になります。これは二度手間です。そこで，第1引数や第2引数がハンドルである場合も，TSMakeEvent関数の第4引数と第5引数には0を指定したほうがよいと思われます。

●相手を指定してメッセージを送る

タスクマンのイベントが作成できたら，それをタスクIDで指定する相手に送信します。それにより，タスクが切り替わり，システムイベントが発生して，タスク間通信が開始されます。

タスクマンのイベントを送信する関数には，

TSCommunicate
TSCommunicateS
TSSendMes
TSPostEventTsk2

の4種類があります。TSCommunicate関数，TSCommunicateS関数，TSSendMes関数はイベントコード13のシステムイベント（ESYSTEM2）を発生し，TSPostEventTsk2関数はイベントコード12のシステムイベント（ESYSTEM1）を発生します。機能的には前の3つは，ほぼ同じものです。

TSCommunicate関数，TSCommunicateS関数，TSSendMes関数は，相手のタスクIDと転送するタスクマンのイベントを引数としてシステムイベントを発生させます。これらは，基本的には，相手からの返事を期待する関数ですが，TSCommunicate(S)関数は第3引数で返事を受け付けるかどうかを指定することができます。

なお，相手からの返事があったかどうかは関数の戻り値で知ることができます。これらの関数の戻り値は次のようになっています。

0	正常終了（返事はない）
2	返事があった
－1	相手が存在しない/別のタスクと通信中
－2	相手の準備ができていない

SX本やシャープからのドキュメントではTSCommunicate(S)関数で返事があった場合の戻り値は1になっていますが，少なくともSX-WINDOWのver2.00以上では2になっているようです。なお，戻り値が－2の場合は，相手の準備ができてない状態ですから，それ以外の戻り値が返るまではメッセージの転送を繰り返し行わなければなりません。

メッセージに対する返事がある場合，相手からの返事は，メッセージを送った側の関数の引数で指定したイベントレコードの中に返事が入っています。メッセージを送った側はイベントレコードの中からwhomなりwhom2なりの値を取り出して，相手からの返事を知ることができるのです。このように，タスク間通信ではメッセージもそれに対する返事もすべてタスクマンのイベントというかたちで行われます。

とはいえ，メッセージに対する返事は，引数で指定したイベントレコードのアドレスを介して受け渡されているのにすぎませんから，そのイベントレコードの内容（ポインタの指す先）はイベントレコードの構造をしている必要はありません。極端な話，文字列でもかまわないのです（この場合は長さが，イベントレコードの大きさであるsozeof(tsevent)より短くなければならないという制限を受けますが)。返事がイベントレコードの構造をしているかどうかをチェックするものはなにもありません。

さて，最後のTSPostEventTsk2関数はほかの3つの関数とは機能が少し異なっています。これは，相手のタスクに対してイベントを発行するだけです。メッセージを送ったあとは知らんぷりで，それに対する返事は期待しません。あとはメッセージを受け取った側がそれに対してなんらかの動作をしてくれるのを期待するだけです。

また，TSPostEventTsk2関数は引数の形式もほかの3つの関数とは異なっています。この関数は相手のタスクIDに加え，TSMakeEvent関数と同様のイベントを作るための情報を引数とします。これらの情報を元にTSPostEventTsk2関数は自分でタスクマンのイベントを作り出してメッセージ転送を行います。いうなれば，TSMakeEvent関数とTSSendMes関数が合体したものというところでしょうか。

TSPostEventTsk2関数はTSMakeEvent関数と同じく，第1引数，第2引数，第3引数が，それぞれ，イベントレコードのwhom，whom2，what2に対応し，第4引数と第5引数で，第1引数と第2引数がハンドルなら，そのハンドルを別の領域にコピーして使用するかどうかを指定します。TSPostEventTsk2関数において，第4引数や第5引数の指定によって新たに生成されたハンドルは，タスクマンがそのイベントレコードを廃棄する際に自動的に廃棄されます。

●相手を指定せずにメッセージを送る

これは，存在するすべてのタスクにメッセージを転送する場合です。このためには，

TSPostEventTsk

という関数を使用します。これは，相手のタスクIDを指定しないTSPostEventTsk2関数と思ってかまいません。相手のタスクIDがない以外は，引数はすべてTSPostEventTsk2関数と同じものです。当然，メッセージを受け取った側からの返事は期待しません。

●メッセージに対して返事をする

タスク間通信のメッセージに対して返事をする場合は，

　　TSAnswer

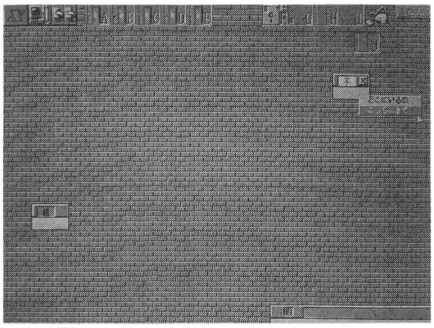
という関数を使用します。システムイベントの12(E_SYSTEM1)あるいは13(E_SYSTEM2)が発生した直後(TSEventAvail関数によって次のイベントを得るまでのあいだ)に，返事とすべきイベントレコードのポインタを引数としてこの関数を呼び出します。それにより，メッセージを送った相手に返事をすることができます。TSAnswer関数の引数に指定されたイベントレコードの内容は，TSCommunicate(S)関数やTSSendMes関数の引数に指定したイベントレコード（元のメッセージ）と同じ場所にそのままコピーされて引き渡されます。

サンプルプログラム

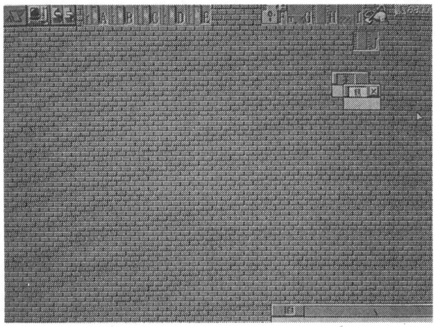
タスク間通信を行う関数の概要がわかったところで，実際にそれらを使ったサンプルプログラムを作ってみましょう。

プログラムは，「親」と「子」を示す２つのタスク（ウィンドウ）が，お互いのウィンドウが存在するグローバル座標をタスク間通信でやりとりしながら，片方が他方の傍に移動する，という内容にしましょう。

このとき，「子」のタスクがいつも主体になってメッセージを転送するものとします。本来なら，このようなメッセージの転送は標準ボタンなどで起動するのが筋でしょうが，ここではプログラムの行数を節約するためポップアップメニューによってメッセージの転送を開始するようにしておきます。いまはメッセージとして次の２種類を考えます。



この位置から



ウィンドウが飛んでくる

ひとつは「親」に対して，「親」のウィンドウの位置のグローバル座標を教えてもらうことを要求するメッセージです。もうひとつは「親」に対して自分（「子」）の傍まで移動してくることを要求するメッセージです。

どちらの場合もタスクマンのイベントの種類（what2）は300ということにしておき，引数(whom)で２種類のメッセージを区別するようにします。すなわち，ひとつ目は，

　　“Catch me.”

という文字列（を示すハンドル），２つ目は，

　　“Come here.”

という文字列（を示すハンドル）を引数として転送することにします。しかし，これだけの情報では「親」が「子」の傍に移動してくることはできないので，もうひとつの引数(whom2)として，「子」のウィンドウのグローバル座標を同時に転送することにします。

なお，これらのメッセージは，ポップアップメニューでは，

　　“どこにいるの”

　　　　　　(“Catch me.”に対応)

　　“こっちにきて”

　　　　　　(“Come here.”に対応)

というメニュー項目で選択するものとします。

さて，「子」はメッセージを転送したあとに２つの動作をします。“Catch me.”に対しては，親からグローバル座標の返事が返るのを待ち，その位置にWMMove関数で自分（「子」）のウィンドウを移動します。“Come here.”に対しては，それ以上なにもせず，「親」のウィンドウが自分に寄ってくるのを待ちます。これが，今回のタスク間通信における「子」の動作です。

「親」のウィンドウもメッセージの種類によって２つの動作をします。“Catch me.”が転送されてきた場合は，自分（「親」）のウィンドウの位置のグローバル座標を返事として返します。“Come here.”が転送されてきた場合は，第２引数（whom2）の位置にWMMove関数で自分のウィンドウを移動します。これが「親」の動作です。

そして，以上のようなタスク間通信を実現するためのプログラムをリスト１(「子」)とリスト２(「親」)に示します。なお，「親」のタスクＩＤをTSFindTskn関数で名前を指定して検索する都合上，リスト２をコンパイルしてできる実行形式のファイル名はtskmaster.xにしておいてください。それが嫌な人はリスト１のFindMasterという関数の中にある「親」タスクの名前を適当に変更してくださいね。

ところで，リスト１とリスト２では，自分自身のウィンドウをオープンする前に，TSFindOwn関数を呼んで，同じ名前のタスクが２つ以上起動されないようにチェックを入れてあります。しかし，これにはそれほど深い意味はありません。

もし，「子」や「親」が複数存在している場合は，「子」から「親」への通信はTSFindTskn関数で最初に見つかった「親」のタスクに対して行われます。余力のある人は，複数存在するタスクがすべて，メッセージを送ったタスクの周りに移動してくるようなプログラムに書き換えてみても面白いでしょう。

おわりに

今回はタスク間通信の基本を説明してみました。創刊10周年PRO-68Kにひっそりと収録されていたsxplay.xのダイアログに「SX-BASICからのコマンド」というのがあって，なんだろうと疑問を持った人がいるかもしれません。あれは，今後発表されるであろうSX-BASIC（作者は石上達也氏）から，タスク間通信によってほかのタスクを制御する試みに対処してみたものです。

SX-BASICはまだ発表されてはいませんが，普通のプログラムからも，what2が258で，whomがコマンドを格納した文字列のハンドルであるようなメッセージを送ることによってsxplay.xを制御することができます。興味を持った人は試してみてはいかがでしょう。

さて，次回はなにをやるか，まだ決めていません。テキストマンにするか，タスクマンの残りの関数にするか，選択肢はいろいろあります。なににするか，もう少し考えさせてください。それでは次回までさようなら。

≪参考文献≫

1）吉沢正敏，SX-WINDOWプログラミング，ソフトバンク，1991年.

2）吉沢正敏，追補版SX-WINDOWプログラミング，ソフトバンク，1991年.

リスト1

```c
/*

     SX-WINDOW  タスク間通信の送り手(子)

                                        (C) 中森 章, Sep.15, 1992
*/
#include <stdio.h>
#define __POINT_T   /* point_t 型を使う */
#include <sxlib.h>
#define FALSE   0
#define TRUE    ~FALSE

/*
     ここでウィンドウに関する定数を設定
*/
#define WDEFID       49
#define WINOPT       0
#define WINWIDTH     64
#define WINHIGHT     18
#define WINTITLE     "¥004 子 "
#define EVENTMASK    EM_EVERY
/*
     ここは定数から計算される定数
*/
#define WINOPTL      ( WINOPT & 0xf )
#define WINDEFID     ( WDEFID << 4 | WINOPTL )

#define MESSAGE      300

window  *winPtr;
rect    winSize;
event   eventRec;
int activeFlag;
int ctrlFlag;   /* コントロールがあるかないか */

#ifdef __GNUC__
asm( ".xdef _STACK_SIZE"  );
asm( "_STACK_SIZE equ   8192" );
asm( ".xdef _HEAP_SIZE"   );
asm( "_HEAP_SIZE equ    16384" );
#endif

main()
{
     if( SX_init()==FALSE ){
          DMError(0x101,"ウィンドウがオープンできません");
          exit();
     }
     while( 1 ){
          TSEventAvail(EVENTMASK,(tsevent*)&eventRec);
          switch( eventRec.eWhat ){
          case E_IDLE:      procIDLE();     break;
          case E_MSLDOWN:   procMSLDOWN();  break;
          case E_MSLUP:     procMSLUP();    break;
          case E_MSRDOWN:   procMSRDOWN();  break;
          case E_MSRUP:     procMSRUP();    break;
          case E_KEYDOWN:   procKEYDOWN();  break;
          case E_KEYUP:     procKEYUP();    break;
          case E_UPDATE:    procUPDATE();   break;
          case E_ACTIVATE:  procACTIVATE();break;
          case E_SYSTEM1:   procSYSTEM();   break;
          case E_SYSTEM2:   procSYSTEM();   break;
          case E_USER1:     procUSER();     break;
          case E_USER2:     procUSER();     break;
          }
     }
}

SX_init()
{
     task    taskBuf;
     short   taskID;

     TSGetTdb(&taskBuf, -1);
     taskID=(short)TSFindOwn();
     if( taskID >= 0 ) {
          DMError( 1, "このプログラムは複数実行できません" );
          exit();
     }
     if( (TSTakeParam(&taskBuf.command,&winSize,NULL,0,NULL,NULL)&1)==0 ){
          *(int *)&winSize.left = TSGetWindowPos();
          winSize.right  = winSize.left + WINWIDTH;
          winSize.bottom = winSize.top  + WINHIGHT;
     }
     winPtr=WMOpen(NULL,&winSize,(LASCII*)WINTITLE,TRUE,WINDEFID,(window *)-
1,TRUE,TSGetID());
     if( winPtr == NULL ) return( FALSE );
     winPtr->wOption = WINOPT;
     activeFlag=FALSE;
     ctrlFlag  = CtrlPrepare();/* コントロールが不要なら ctrlFlag=FALSE */
     drawGrowBox();
     return( TRUE );
}

SX_term()
{
     if( ctrlFlag ) CtrlDispose();
     WMDispose( winPtr );
     exit();
}

drawGrowBox()
{
     GMSetGraph( &winPtr->wGraph );
     WMDrawGBox( winPtr );
}

CtrlPrepare()
{
     return( FALSE );
}

CtrlDispose()
{
     return( FALSE );
}

procIDLE()
{
     return( FALSE );
}

procMSLDOWN()
{
     if( (window*)eventRec.eWhom != winPtr ) return( FALSE );
     if( activeFlag == FALSE ){
          WMSelect( winPtr );
          activeFlag = TRUE;
          if( EMLStill() == 0) goto checkDClick;
     }
     switch( SXCallWindM(winPtr,(tsevent*)&eventRec) ){
     case W_INCLOSE:
          SX_term(); break;
     case W_INGROW:
     case W_INZMOUT:
     case W_INZMIN:
          GMClipRect(&winPtr->wGraph.grRect);
          break;
     }
checkDClick:
     TSGetEvent(EVENTMASK,(tsevent*)&eventRec);
     return( TRUE );
}

procMSLUP()
{
     return( FALSE );
}

procMSRDOWN()
{
     menu    **menuHdl;
     tsevent Event;
     int     item;
     point_t pos;

     if( (window*)eventRec.eWhom != winPtr ) return( FALSE );
     TSGetEvent( EVENTMASK, (tsevent*)&eventRec );
     menuHdl=MNConvert(0,
          "どこにいるの,"
          "こっちにきて"            ,/*MDEFID*/0);
     if(menuHdl<(menu**)0) return;
     item = MNSelect( menuHdl, eventRec.eWhere );
     switch(item){
     case 1:
          if(SendMessage("Catch me.",FindMaster(),&Event)>0){
               pos.x_y=Event.whom;       /* 左上のグローバル座標 */
               pos.p.x+=20;              /* 少し座標をずらす */
               pos.p.y+=20;
               WMMove(winPtr,pos,1);    /* そこに移動する */
          }
          else{
               DMError(1,"返事がありません");
          }
          break;
     case 2:
          SendMessage("Come here.",FindMaster(),&Event);
          break;
     }
}

procMSRUP()
{
     return( FALSE );
}

procKEYDOWN()
{
     return( FALSE );
}

procKEYUP()
{
     return( FALSE );
}

procUPDATE()
{
     if( (window*)eventRec.eWhom != winPtr ) return( FALSE );
     WMUpdate( winPtr );
     if( ctrlFlag ) CMDraw( winPtr );
     WMUpdtOver( winPtr );
     drawGrowBox();
     TSGetEvent(EVENTMASK,(tsevent*)&eventRec);
}

procACTIVATE()
{
     if( (window*)eventRec.eWhom == winPtr )    activeFlag = TRUE;
     else if( eventRec.eWhom != NULL ){
          if( activeFlag ) {
               activeFlag = FALSE;
               TSGetEvent(EVENTMASK,(tsevent*)&eventRec);
          }
     }
     return( TRUE );
}

procSYSTEM()
{
     switch( ((tsevent*)&eventRec)->what2 ){
     case CLOSEALL:
     case ENDTSK:
          SX_term(); break;
     case WINDOWSELECT:
          WMSelect( winPtr ); break;
     }
}

procUSER()
{
     return( FALSE );
}

FindMaster()
{
     return TSFindTskn("tsk_master.x",0);
}

SendMessage(cmd,id,evp)
char    cmd[];
int id;
tsevent *evp;
{
     tsevent Event;
     int slen;
     int stat;
     char     **hdl;
          point_t pos;
          GMSetGraph( &winPtr->wGraph );
          pos.p.x=winPtr->wGraph.grRect.left;
          pos.p.y=winPtr->wGraph.grRect.top;
          pos.x_y=GMLocalToGlobal( pos );
     slen=strlen(cmd)+1;
     hdl=(char**)MMChHdlNew(slen);    /* ハンドルを作る */
     if(hdl==0) return( -1 );
     strcpy(*hdl,cmd);    /* ハンドルにコマンド(メッセージ)をコピー */
     TSMakeEvent((long)hdl, pos.x_y, MESSAGE, 0, 0, &Event);
     do{
```

```
#if 0
    /* 返事が必要でないなら

         TSPostEventTsk2((long)hdl, pos.x_y, MESSAGE, 1, 0, id)

        でも同じこと。ただし、発生するイベントコードが異なる。
    */
         stat=TSCommunicate(id,&Event,1);
#else
         stat=TSSendMes(id,&Event);
#endif
    }while(stat==-2);           /* 相手の準備ができるまで繰り返し */
    if(stat>0){                 /* 返事があったら */
        *evp=Event;             /* 返事を引き取る */
    }
    MMHdlDispose(hdl);          /* ハンドルを廃棄 */
    return( stat );
}
```

リスト2

```
/*

    SX-WINDOW  タスク間通信の相手（親）

                                      (C) 中森 章, Sep.15, 1992
*/
#include <stdio.h>
#define __POINT_T   /* point_t 型を使う */
#include <sxlib.h>
#define FALSE   0
#define TRUE    ~FALSE

/*
    ここでウィンドウに関する定数を設定
*/
#define WDEFID       49
#define WINOPT       0
#define WINWIDTH     64
#define WINHIGHT     18
#define WINTITLE     "¥004 親 "
#define EVENTMASK    EM_EVERY
/*
    ここは定数から計算される定数
*/
#define WINOPTL      ( WINOPT & 0xf )
#define WINDEFID     ( WDEFID << 4 | WINOPTL )

#define MESSAGE      300

window  *winPtr;
rect    winSize;
event   eventRec;
int activeFlag;
int ctrlFlag;   /* コントロールがあるかないか */

#ifdef __GNUC__
asm( ".xdef _STACK_SIZE"  );
asm( "_STACK_SIZE equ   8192" );
asm( ".xdef _HEAP_SIZE"    );
asm( "_HEAP_SIZE equ    16384" );
#endif

main()
{
    if( SX_init()==FALSE ){
        DMError(0x101,"ウィンドウがオープンできません");
        exit();
    }
    while( 1 ){
        TSEventAvail(EVENTMASK,(tsevent*)&eventRec);
        switch( eventRec.eWhat ){
        case E_IDLE:      procIDLE();     break;
        case E_MSLDOWN:   procMSLDOWN(); break;
        case E_MSLUP:     procMSLUP();    break;
        case E_MSRDOWN:   procMSRDOWN(); break;
        case E_MSRUP:     procMSRUP();    break;
        case E_KEYDOWN:   procKEYDOWN(); break;
        case E_KEYUP:     procKEYUP();    break;
        case E_UPDATE:    procUPDATE();   break;
        case E_ACTIVATE:  procACTIVATE();break;
        case E_SYSTEM1:   procSYSTEM();   break;
        case E_SYSTEM2:   procSYSTEM();   break;
        case E_USER1:     procUSER();     break;
        case E_USER2:     procUSER();     break;
        }
    }
}

SX_init()
{
    task    taskBuf;
    short   taskID;

    TSGetTdb(&taskBuf, -1);
    taskID=(short)TSFindOwn();
    if( taskID >= 0 ) {
        DMError( 1, "このプログラムは複数実行できません" );
        exit();
    }
    if( (TSTakeParam(&taskBuf.command,&winSize,NULL,0,NULL,NULL)&1)==0 ){
        *(int *)&winSize.left = TSGetWindowPos();
        winSize.right  = winSize.left + WINWIDTH;
        winSize.bottom = winSize.top  + WINHIGHT;
    }
    winPtr=WMOpen(NULL,&winSize,(LASCII*)WINTITLE,TRUE,WINDEFID,(window *)-
1,TRUE,TSGetID());
    if( winPtr == NULL ) return( FALSE );
    winPtr->wOption = WINOPT;
    activeFlag=FALSE;
    ctrlFlag  = CtrlPrepare();/* コントロールが不要なら ctrlFlag=FALSE */
    drawGrowBox();
    return( TRUE );
}

SX_term()
{
    if( ctrlFlag ) CtrlDispose();
    WMDispose( winPtr );
    exit();
}

drawGrowBox()
{
    GMSetGraph( &winPtr->wGraph );
    WMDrawGBox( winPtr );
}

CtrlPrepare()
{
    return( FALSE );
}

CtrlDispose()
{
    return( FALSE );
}

procIDLE()
{
    return( FALSE );
}

procMSLDOWN()
{
    if( (window*)eventRec.eWhom != winPtr ) return( FALSE );
    if( activeFlag == FALSE ){
        WMSelect( winPtr );
        activeFlag = TRUE;
        if( EMLStill() == 0) goto checkDClick;
    }
    switch( SXCallWindM(winPtr,(tsevent*)&eventRec) ){
    case W_INCLOSE:
        SX_term(); break;
    case W_INGROW:
    case W_INZMOUT:
    case W_INZMIN:
        GMClipRect(&winPtr->wGraph.grRect);
        break;
    }
checkDClick:
    TSGetEvent(EVENTMASK,(tsevent*)&eventRec);
    return( TRUE );
}

procMSLUP()
{
    return( FALSE );
}

procMSRDOWN()
{
    return( FALSE );
}

procMSRUP()
{
    return( FALSE );
}

procKEYDOWN()
{
    return( FALSE );
}

procKEYUP()
{
    return( FALSE );
}

procUPDATE()
{
    if( (window*)eventRec.eWhom != winPtr ) return( FALSE );
    WMUpdate( winPtr );
    if( ctrlFlag ) CMDraw( winPtr );
    WMUpdtOver( winPtr );
    drawGrowBox();
    TSGetEvent(EVENTMASK,(tsevent*)&eventRec);
}

procACTIVATE()
{
    if( (window*)eventRec.eWhom == winPtr )    activeFlag = TRUE;
    else if( eventRec.eWhom != NULL ){
        if( activeFlag ) {
            activeFlag = FALSE;
            TSGetEvent(EVENTMASK,(tsevent*)&eventRec);
        }
    }
    return( TRUE );
}

procSYSTEM()
{
    char    **msg;
    char    cmd[128];
    tsevent Event;
    point_t pos;
    int     stat;

    switch( ((tsevent*)&eventRec)->what2 ){
    case CLOSEALL:
    case ENDTSK:
        SX_term(); break;
    case WINDOWSELECT:
        WMSelect( winPtr ); break;
    case MESSAGE:
        msg=(char**)((tsevent*)&eventRec)->whom; /* 引数を取り出す */
        if(strcmp("Catch me.",*msg)==0){
            GMSetGraph( &winPtr->wGraph );
            pos.p.x=winPtr->wGraph.grRect.left;
            pos.p.y=winPtr->wGraph.grRect.top;
            pos.x_y=GMLocalToGlobal( pos ); /* 自分の左上のグローバル座標を得る */
            TSMakeEvent(pos.x_y, 0, MESSAGE, 0, 0, &Event);
            do
                stat=TSAnswer( &Event );    /* 自分のグローバル座標を返す */
            while( stat!= -2);
        }
        else if(strcmp("Come here.",*msg)==0){
            pos.x_y=((tsevent*)&eventRec)->whom2;    /* 相手のグローバル座標 */
            pos.p.x+=20;                             /* 少し座標をずらす */
            pos.p.y+=20;
            WMMove(winPtr,pos,1);                    /* そこに移動する */
        }
        break;
    }
}

procUSER()
{
    return( FALSE );
}
```

コンピュータアーキテクチャ編

外部電源の製作

Misawa Kazuhiko
三沢 和彦

電子回路の理論と実習が終わって，今回は電源回路の解説，汎用性のある電子回路用の電源を製作します。ハードウェア工作のためだけでなく，自分の目的に合った電源を設計したいときに役立つことでしょう。

先月は２桁の繰り上がり付き加算器を製作しました。TTL IC１個だけの回路なので，わりと簡単に工作できたかとも思いますが，いかがでしたか。今月は分量の都合で先月中に説明しきれなかった，専用電源の製作を説明していきたいと思います。

この連載はデジタル論理回路の設計製作がテーマとなっているので，電源回路そのものは対象外かもしれません。しかしながら，これから連載を続けていくにあたって，ほとんどすべての回路に＋5V電源が必要となってきます。

そのたびにX68000から電源を引っ張ってくるのは，電源として利用するためだけにX68000の電源を入れて，かつジョイスティックポートを占有しなければならないので，非常に不便です。一度作ってしまえばいいのですから，いまのうちから専用電源を作ってしまうというのも手です。

また，X68000を所有していない読者もいるでしょうから，ここから作り始めるのもハードウェア工作の連載としては正しいあり方かもしれません。

そこで，IC１個程度でできる簡単なデジタル回路専用電源を設計してみました。使用するICは３端子レギュレータというもので，電源回路によく使われるICです。回路図だけは，先月のうちに出しておきましたが，それだけで実際に製作するのは難しいかもしれません。今月は改めて，回路の仕組みやそれぞれの部品の役割などから丁寧に解説していきたいと思います。

直流電源とは

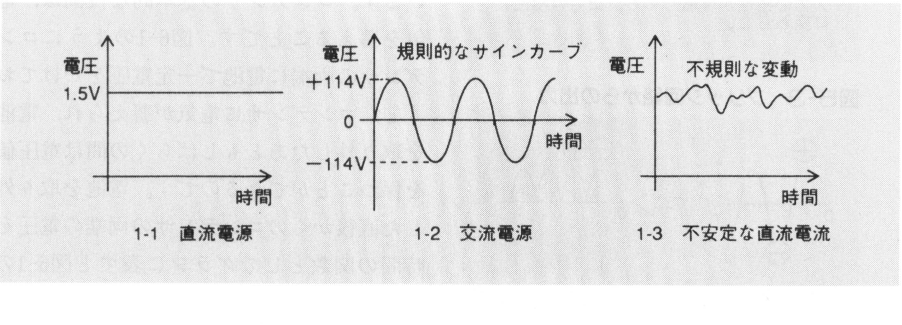
電源には大きく分けて直流と交流の２種類があります。直流というのは，出力電圧が時間にかかわらず一定値を取るものです。交流というのは出力が時間と共に規則正しく変動しているものです。

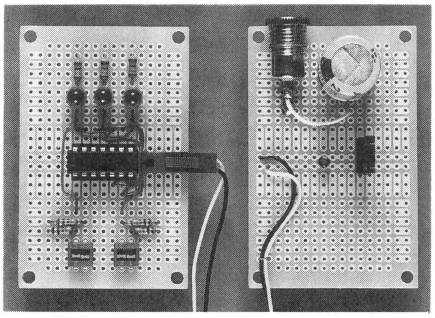
電池などは，図1-1のように常に1.5Vならその出力電圧を出し続けているので，直流電源ということになります。一方，家庭用コンセントの場合，100Vならば図1-2のように＋114Vから－114Vまでサインカーブを描いて電圧が時間変化しているので，交流電源ということになります。

また，直流電源といえども，図1-3のように出力が不規則に変動している場合もあるので注意が必要です。この場合は，不安定な直流ということになります。

デジタル回路は電圧値のH/Lで演算を行いますので，デジタル回路の電源を考える場合，基準となる電源電圧が時間と共に変動しているようでは困ります。TTL ICでは電源電圧5Vに対して約1Vを境にHとLとが入れ替わりますが，電源電圧が変動すると出力電圧も変動してしまいます。

図１　電源の出力電圧




外部電源をつないだ加算器回路

もし，H/L境界あたりの1V付近で変動していると，同じ入力に対して，あるときはHと判断されても別のときにLと判断されてしまうのです。これでは，正確な動作は望めません。そこで，出力電圧が一定している電源を用意しなければならないのです。

出力が変動している電圧値を安定させる回路をレギュレータといい，そのレギュレータを使った電源を安定化電源といいます。安定化電源には，出力電圧を自由に設定できる可変型安定化電源と，ある特定の電圧値しか取り出せない定電圧電源とがあります。可変型はボリュームで目的の電圧値に設定すれば，あとはその値で一定値を出力します。ICによっては，電源電圧が5V，6V，あるいは12Vが必要なものもあるため，いろいろな用途に使うには，可変型安定化電源が便利です。

しかし，TTL ICを使った回路に限定して使うとすれば，電源電圧が＋5Vと決まっていますので，出力電圧が＋5Vに固定化されていてもよいことになります。そのような定電圧電源を作るには，これから説明する３端子レギュレータというICがもっとも手頃で便利な部品です。

安定化電源の構成

メインとなる3端子レギュレータについて詳しく説明する前に，安定化電源回路の全体像を把握しておきましょう。交流電圧を入力として，電圧変換，整流，平滑，安定化の各ブロックを経て直流安定電圧が出力されます（図2）。

1） 電圧変換

家庭用コンセントは100Vですが，必要な直流電圧が5Vの場合には，100Vから電圧を落とさなければなりません。これには，トランスというものを使います。

トランスとは，電線を鉄の芯にぐるぐると何重にも巻き付けたコイルを，2つひと組で向かい合わせに組み合わせたものです。この2つのコイルの片方に100Vの交流をかけると，電磁誘導という仕組みによって，もう片方のコイルにも交流が生じます（図3）。なお，電磁誘導というのは，片方のコイルに電流を流したとき，電流によって発生する磁力線がもう片方のコイルに作用し，それによって電圧が生じる現象のことをいいます。

また，新たに発生する交流電圧はこれらのコイルの電線の巻き数によって決まります。適切なトランスを使えば，100Vの交流を任意の電圧の交流に変換することができるのです。いま，5Vの直流電圧を得ようとすると，あとで述べるように9V程度の入力電圧が必要になるので，ここではトランスによって9Vに変換します。

2） 整流

交流電圧というのは，時間と共に正負が逆転しています。ですから，そのまま回路に電圧をかけると，ある瞬間に＋端子へ－電圧がかかってしまうことになり，回路を壊してしまいます。

そこで，常に電圧の正負が変わらないように，極性を揃えなければなりません。これには，ダイオードという素子を使います。ダイオードには2つの端子があって，基本的には片方の向きの電流しか流さない働きがあります。

つまり，ダイオードの両端に正負の入れ替わる交流電圧をかけると，電流が流れるときと流れないときとが交互に入れ替わっている状態になります(図4)。しかし，これでは電圧がかかっているときといないときがあることになり，一定電圧をかけることにはなりません。

そこで，安定化電源に使われる整流回路では，このダイオードを図5-1のように4本組み合わせた，ブリッジ回路が一般的に使われています。このブリッジ回路を使うと，電圧の正負が逆転しても，図5-2のように電流の流れるダイオードの組みが入れ替わることになり，出力に変動があるものの，常に同じ向きに電圧がかかるようにすることができるのです。結果的には整流器からは図5-3のような出力が得られることになります。

3） 平滑

図5に示したようにただ整流しただけでは，電圧値の正負が揃っただけで，電圧値そのものの大きさはまだ変動しています。

そこで，この変動を滑らかにする，つまり電圧の値全体を一定値に揃える平滑という作業が必要になってきます。

この平滑をするためにはコンデンサを使います。コンデンサの基本的な役割は，電気を蓄えることです。図6-1のようにコンデンサの両端に電池で一定電圧をかけておくと，コンデンサに電気が蓄えられ，電池を取り外したあともしばらくの間は電圧値を保つことができるのです。電池を取り外した直後からのコンデンサの両端の電圧を時間の関数としてグラフに表すと図6-1の

図2　安定化定電圧電源の構成（100V交流→5V直流の場合）

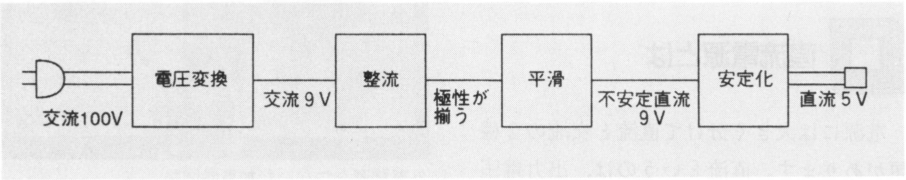


図3　トランスの仕組み

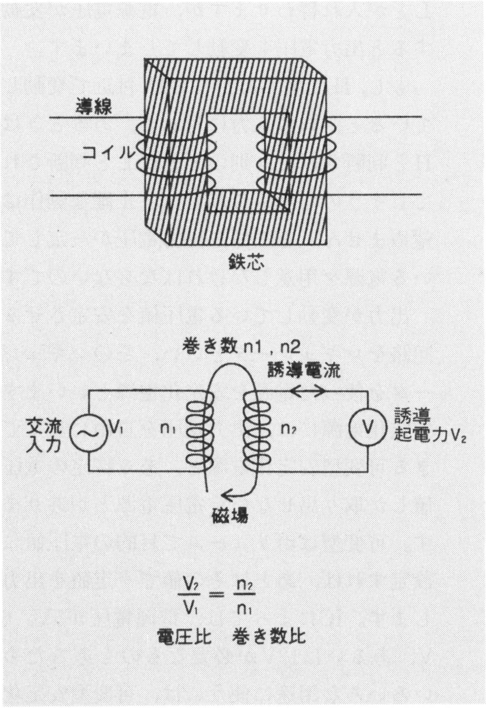


図4　ダイオードの仕組み

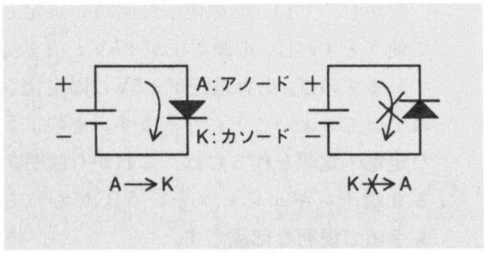


図5-1　ブリッジ回路

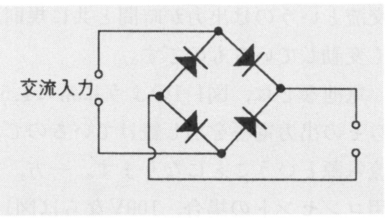


図5-2　ブリッジ回路からの動作

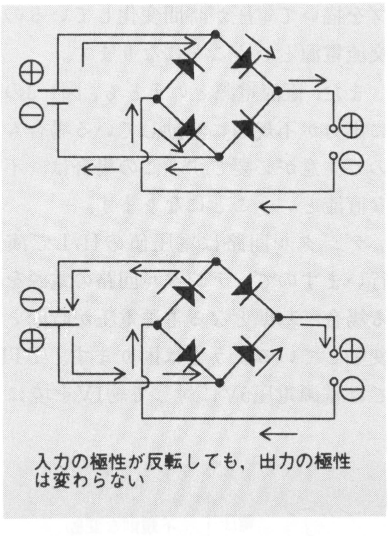
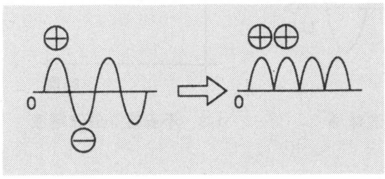


図5-3　ブリッジ回路からの出力



ようになります。

このコンデンサに変動した電圧がかかるとどうなるでしょうか？　結果は図6-2のように波形がなまってかなり一定値に近づいてきます。

4)　安定化

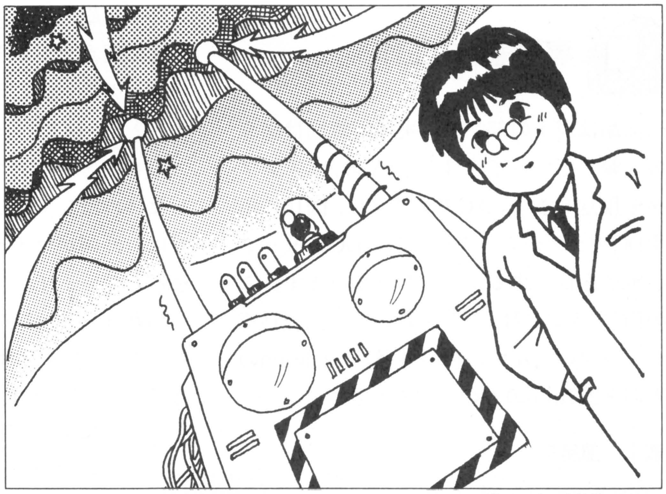
コンデンサで平滑しただけでは，完全に安定した電圧にはならず，図6-2のようにリップルという小さな変動がのっています。最後にこのリップルを除くため，レギュレータICに通して安定化します。

3端子レギュレータの使い方

3端子レギュレータには，代表的なものとして正電圧用の78シリーズと負電圧用の79シリーズとが挙げられます。このシリーズは出力電圧，出力電流に対応した多くの品種が揃っているのが特徴的です。出力電圧には+/−5，8，12，15，18，20，24Vなどがあり，出力電流には100mA，500mA，1Aなどがあります。

今回TTL IC回路用の+5V電源として製作する回路は図7のようなものです。回路図は先月と同じものですが，参考までにACアダプタの中身の回路図も書き下しています。回路図を見ればわかると思いますが，ACアダプタというものは電圧変換のトランス，整流用のブリッジ，平滑用のコンデンサがひとつのケースに入っているだけのものなのです。

このACアダプタを使って家庭用コンセントの100Vから9Vに電圧変換します。最終的に5Vが必要な場合は3端子レギュレータ7805に入力する交流電圧は7V以上必要なので，一般に手に入れやすい9Vタイプを選びました。単純に考えると5VのACアダプタを使えば，直接TTL IC用の電源が得られるように思うかもしれません。しかし，ACアダプタの出力というのは平滑までしか行っていないので，リップルという変動成分がのっているために，きちんと一定の電圧を出していないのです。



illustration：Y.Kawahara

つまり，ACアダプタは交流を直流に変換するものの，定電圧電源というわけではないのです。しかしながら，定電圧電源の基本構成から考えると残っているのは安定化だけですから，レギュレータを通してやりさえすれば，TTL ICを駆動するのに十分な電源回路が出来上がるというわけです。

ところで，レギュレータICに必要な入力は5V出力で7Vと書きましたが，これはどういうことでしょうか。もし，平滑後の入力が図8のようにリップルが大きいために一部分7Vを割っている場合には，7Vより低い分はリップルが除去されないで，図8の下の波形のようになってしまうのです。

図6-1　平滑コンデンサ

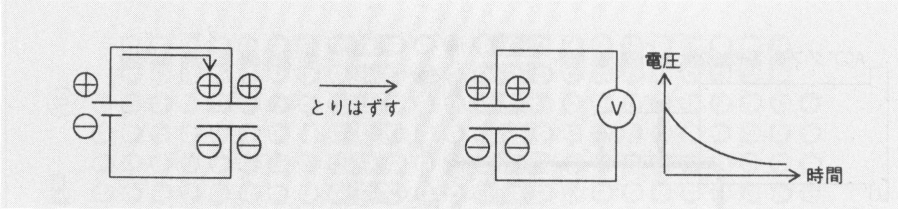


図6-2　平滑後の波形

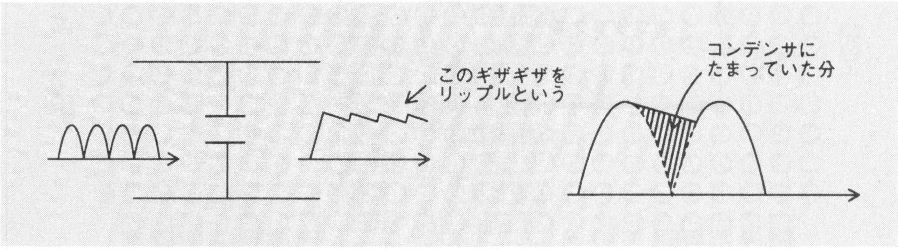


図7　回路図

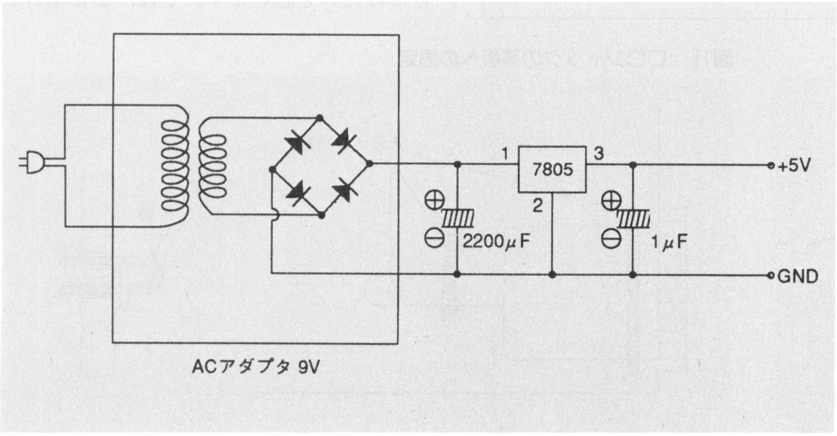
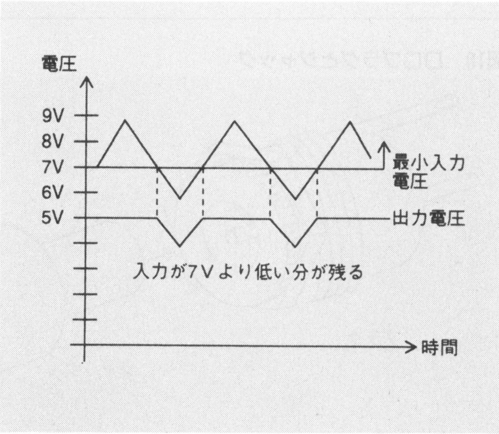


図8



専用電源の工作

部品は表1の部品表，実体配線図は図9のとおりです。工作はまず9VのACアダプタを用意します。ACアダプタは，図10に示すDCプラグから，受ける側のDCジャックに電源を供給するようになっています。このDCプラグには，差し込みの穴の内径(DCジャックの真ん中の棒の外径) が2.5mmのものと2.1mmのものとがあります。

表1　部品表

IC用基板（サンハヤトICB-87）	1枚	90円
IL2ピンコネクタオス（日本航空電子JAE）	1個	40円
コンタクトピン	1組	80円
3端子レギュレータ7805	1個	80円
2200μF電解コンデンサ	1個	100円
1μF電解コンデンサ	1個	25円
DCジャック（2.5φ，2.1φ）	各1個	@100円
スズメッキ線	少々	
ビニール配線材	少々	

部品を購入する際に，DCジャックとして2種類用意しておき，購入したACアダプタのプラグの太さに合っているほうを使います。また，DCプラグの極性は，外側がプラスのものとマイナスのものとがあり，注意しなければなりません。今月の実体配線図では，外側がマイナスのものを使った場合を示しています。

最初は，DCジャックを基板に取り付けるところから始めます。スズメッキ線を1cmほどに切って，図11のようにDCジャックの端子に通して基板の穴に差し込み，ハンダ付けします。それだけではぐらぐらしてしまいますので，スズメッキ線を本来ねじ留めする本体に巻き付けてハンダ付けします。DCジャックの真ん中の端子のほうは，ビニール被覆線で配線する以外にすることはありません。

DCジャックが基板に取り付けられれば，あとは簡単です。次に電解コンデンサを取り付けます。電解コンデンサは，小型で容量を大きくするために容器内に電解液を満たしたタイプのコンデンサで，足に正負の極性があり，本体表面にマイナス記号が付いているので，間違えずに配線してください。電解コンデンサはレギュレータICの入力側と出力側とに2個あります。

最後に3端子レギュレータICそのものをハンダ付けして，基板は完成です。レギュレータICの足の順番も間違えないようにしましょう。型番が書いてある表から見て，左から入力，GND，出力の順番です。

この専用電源の出力は，先月述べたように2ピンのコネクタに統一します。コネクタプラグへのハンダ付けは先月号の説明を見てください。

これだけ簡単な回路なので，動作チェックをするほどでもないでしょう。最終的に出力があるかないか2つにひとつだと思いますので，出力がなかったときのチェック事項を挙げておきます。

まず決定的なのは，レギュレータICの足を間違えているケースです。次に電解コンデンサの極性を間違えたためにコンデンサが壊れていることも考えられます。また，ACアダプタからの出力のDCプラグをDCジャックで受けるときにその極性を間違えている場合もあります。あらかじめ，テスターなどでDCプラグの極性を調べておくと間違いありません。

＊　＊　＊

今月は専用電源の製作で手短にまとめてみました。さっそく完成させて，来月以降の論理回路に備えてください。さて来月からは，先月の加算器を発展させて，レジスタとバスの概念を取り入れた加算器に仕上げてみたいと思います。では，また来月。

図9

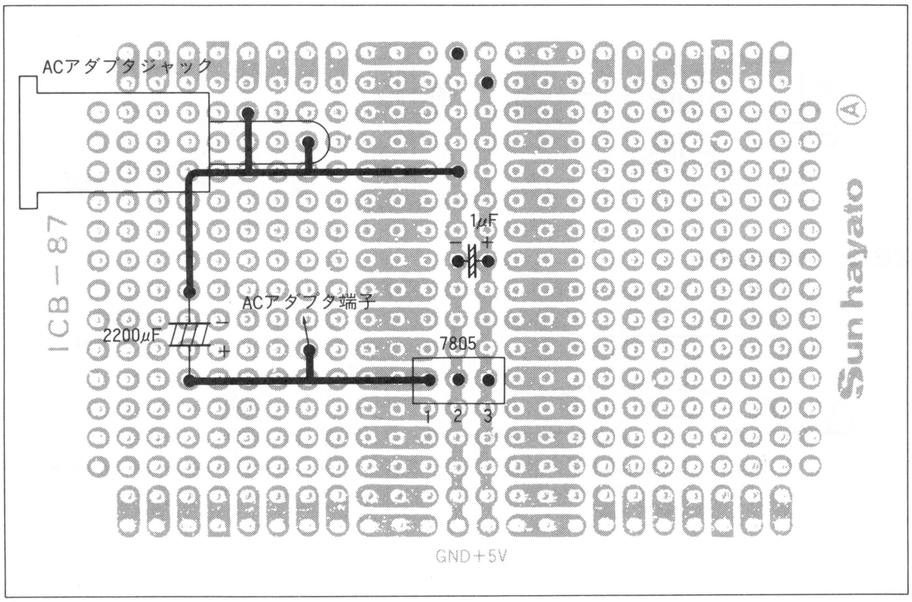


図10　DCプラグとジャック

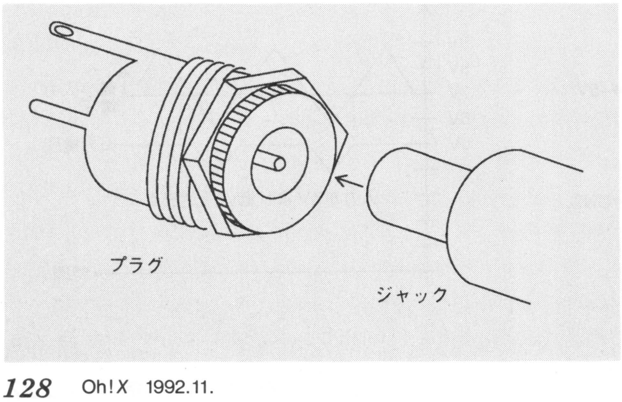
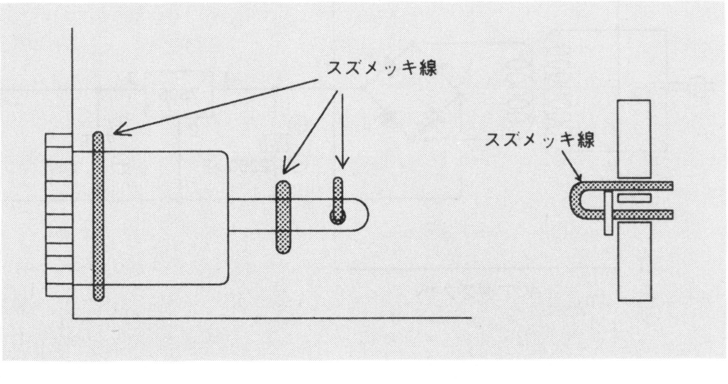


図11　DCジャックの基板への固定



CREATIVE COMPUTER MUSIC

Creative Computer Music入門(14)

分厚いハーモニーを作る

応用編2回目の今回は，緻密で美しく，厚みのあるハーモニーを演出するためのテクニックについて，理論的に説明してみましょう。大切なのは，ハーモニーを構成するのに必要な音を知ることですが，これは，ダウンサイジングのアレンジにも応用できるテクニックです。

Taki Yasushi **瀧　康史**

§ いやがらせは好きですか?

どぉも。最近ハードディスクが壊れてしまって，さめざめと泣いている瀧です。うるる～，これは寂しすぎる……。

と，悲嘆にくれててもしようがないので，まずはCDの紹介をしましょう。今月はいろいろ買ってしまいました。CDだけで相当な数になったかもしれない。そんななかからお気に入りで，今でも聴いているやつ(しまった！そうしたら工藤静香になってしまう)……ではなくて，私はマゾなので(大嘘)いやがらせのCDを聴きましょう。

今回紹介するCDは，7月号のLIVE in'92，そして，8月号のこの連載のコラムでもお話ししたあの「ヴェクサシオン」です。恋は盲目というか(おい，意味が違うぞ！)，燈台下暗しというか，ヴェクサシオンのCDはなんとそこらのCD屋さんでよく売っている例の，「大好きシリーズ」に入っていたのです。といってももちろん部分だけですけどね。CDの名称は「サティ大好き(PHILIPS 20CD3228)」です。

さて，このサティですが，どうも変な人で(私は変な人が好き！)なかなか面白い曲を作ります。人それぞれの評価がありますが，私が感じたのは「表情のなさ」です。といっても，コンピュータミュージックのような，まるっきりの表情のなさではなく，人の言葉を借りれば「クール」だとか「硬質」だとかいろいろありますが，私にいわせればサティの曲は「白い」のです。白いから透明感があって，こう，なんていうのか理屈ではなく，素直に心にじかに染みわたる感じがするのです。また演奏者が誰であるかにかかわらず，ピアノの音の重さが感じられたりもします。なかなか珍しいタイプの曲なのです。

これは，ぜひ聴いてほしいですね。

ところで，このCDでは作者の意図を考えてか，遅く悲痛に演奏されていたりするので，ひょっとしたらもう1枚買うはめになるかもしれません。ちなみに，もっとしっかり探せばあるんでしょうけど，なかなかヴェクサシオンの入っているCDは見つかりませんので……。

それでは，今月の分を始めましょうか。

§ 大事な音は

曲のハーモニーをうまく作り上げようとするとき，ハーモニーを構成するのに最低限必要な音が何なのかを知れば，かなり有利に進めることができるのは明確な事実です。たとえその作業がダウンサイジング的なものであってもです。

たとえば仮に，あなたが同人ソフトの音楽担当者だったとします。ゲームのまとめ役の人もしくは音源ドライバ作成の担当者からは，「MIDI対応だから思いっきりカッコよくして!」と言われたとしましょう。そこで，あなたは気合を入れて，一度に20音ぐらいを使った分厚い曲を作ってみました。ところが，いざ曲を作ってみると，今回はMIDI対応の音源ドライバができないとか，そのほかの理由があって，その曲を再度FM音源で仕上げ直してくれと要望が出たのです。

さて，どうしましょう?

実際に曲を作るときには，こういう問題は結構出てくると思います。

同じようなダウンサイジングのアレンジの例としては，ツインギターのバンドから作曲を頼まれて，いざ曲を作ったらギタリストが1人抜けていたとか，学祭はもう近いのに時間がないとか……。考えてみればダウンサイジングのアレンジは，いろんなところで必要になる場合があるような気がします。

このようなダウンサイジングのアレンジは，たいていは周囲の事情によるものなので，曲を作る側としては少々寂しい気もしてつまらない(実際，やっていてあまり面白いものではない)のですが，ただ曲を作って遊ぶだけならともかく，それで何かをしようとなると，なかなかシビアに問題はかかわり合ってきます。

さて，音を省略するとなると，省略の仕方によってはもともとのアンサンブルが著しく崩れてしまいます。アンサンブルをできるだけ崩さずに音をはしょるには，聴きながらやっていくのがいちばんよいのでしょうが，そういう才能があるならともかく，慣れないと，試行錯誤の果てにめちゃめちゃになってしまいがちです。

夜が明けるまでちょっと直しては聴き，またちょっと直しては聴き，夜明けを見ながら作曲，なんてのもなかなかオツなんですけれど，困ったことに，人間の耳には「慣れ」というものがあります。耳が慣れてくると，バランスがとれているのか，いないのかわからなくなってしまうことがあります。

どうすればいいでしょうか。

結論からいってしまえば，重要な音，すなわち，なくなってしまうとアンサンブルが変わってしまうような音を意識することが重要だということです。

また，今までにあげた例とは逆に，曲に厚みをつけたい場合にはどうすればいいでしょうか。メロディとベースだけはわかっているんだけど，中間にどのような音を加えればよいかわからないときは，どうしましょうか?

感性で，「これとこれとこれが美しいハーモニーを作る！」とわかってしまう人には，関係ないでしょう。また，とりあえず，適当に曲を作ってみたいというだけの人にも，今回はあまり関係がありません。

今回は，緻密で美しく，厚みのあるハーモニーを曲の中に演出したい人に，これらのテクニックを理詰めで話してみましょう。

すでに気づいているでしょうが，これにはコードの知識がものをいいます。

うまく説明逃れしてしまえば，「～っぽいフレーズは」とか「曲によく影響をもたらしている」だとか，曖昧な言い方で(この程度の曲なら)言い逃れできるのですが，

図1　4声体

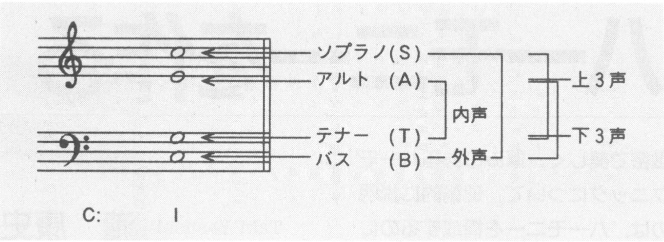


図3　構成音の重複

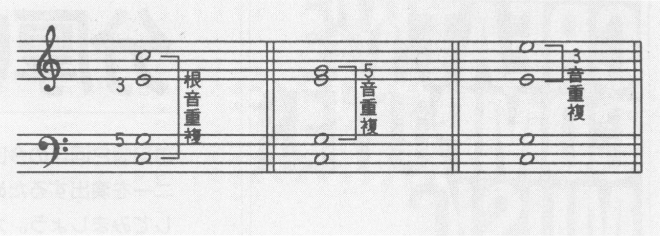
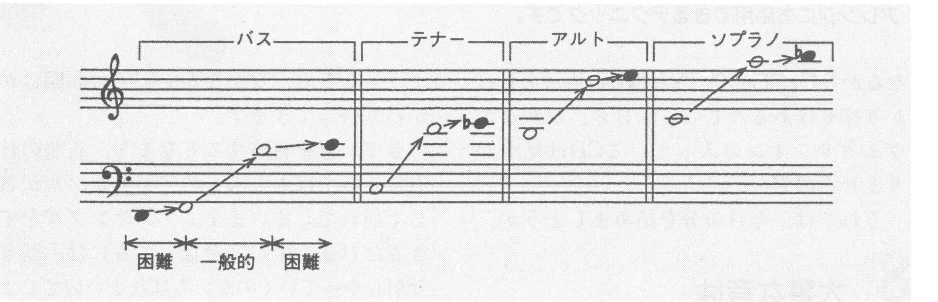


図2　各声部の音域



曲調が変わったり，メロディが絡み合ったりしてくると，まったく応用がききません。結局はその曲でもまた，別の「～っぽい感じ」などで言い逃れをしなくてはならなくなってしまいます。そこで，ここはみっちりコード，またコードから応用される知識をフルに使い，テクを身につけましょう。

§ コードの配置と4声体

しかし，毎回このテクニックを使うたびに思うのですが，こういう理論を考えた人ってすごいですよね。この理論のおかげで，耳に自信がない人でも，知識さえ身につけておけば，まともにアンサンブルがとれるのです。

でも，やはりそれなりの代償はあります。覚えるのが面倒ですし，それよりも，わかりやすく教えるのはかなり大変です。そんな理由で，すぐに役に立つのは知っていながら，今まで説明するのを避けていたのです。しかもこの説明は，難しい音大などから出版されている教科書などにしか載っていなくて，当然ながらとてもじゃないけど一般的ではありません。

それで私のように，和声をくだいて説明してある音楽雑誌なんかを見て，教え方をちょっと拝借しようと思っても……そこでもあまり触れられていません。

というわけで，非常に面倒なのですが，ここは覚悟して覚えてください。理解してしまえば，今後，曲を作るうえでとっても役に立ちます。

さて。まず楽曲を突き詰めて重要な音を4つにまで絞るとします。この4つという音の数は，先月の話にあったとおり，人間の音楽が歌から始まったことに起因します(和声という言葉からも推測できますが)。

人の声には大きく分けて2つ，男性の声と，女性の声があります。また，同性の声でも，声の高い人と低い人ではだいたい4～5度の音域の差があります。そういったわけで4つなのです。昔はこれ以上の多声体もあったのですが，最も落ち着く形として4つに割り振られた，とでもいっておきましょうか。

この形を4声体といいます。

そしてこの4つの音ですが，それぞれに名称があって，ソプラノ，アルト，テナー(テノール)，バス……よく聞きますよね(図1)。これらは基本的には，高い音から順に，ソプラノ，アルト，テナー，バスとなります。女性の高い声がソプラノ，同じく女性の低い声がアルト，男性の高い声がテナー，そして低い声がバスです。

説明の都合上，覚えてもらったほうがよいのでお話ししておきますが，この上下の2つ，すなわちソプラノとバスをまとめて外声，アルトとテナーをまとめて内声といいます。また，上の3つ（ソプラノ，アルト，テナー）を上3声(じょう3せい)，下

楽譜1　バナナパフェ味のそよ風

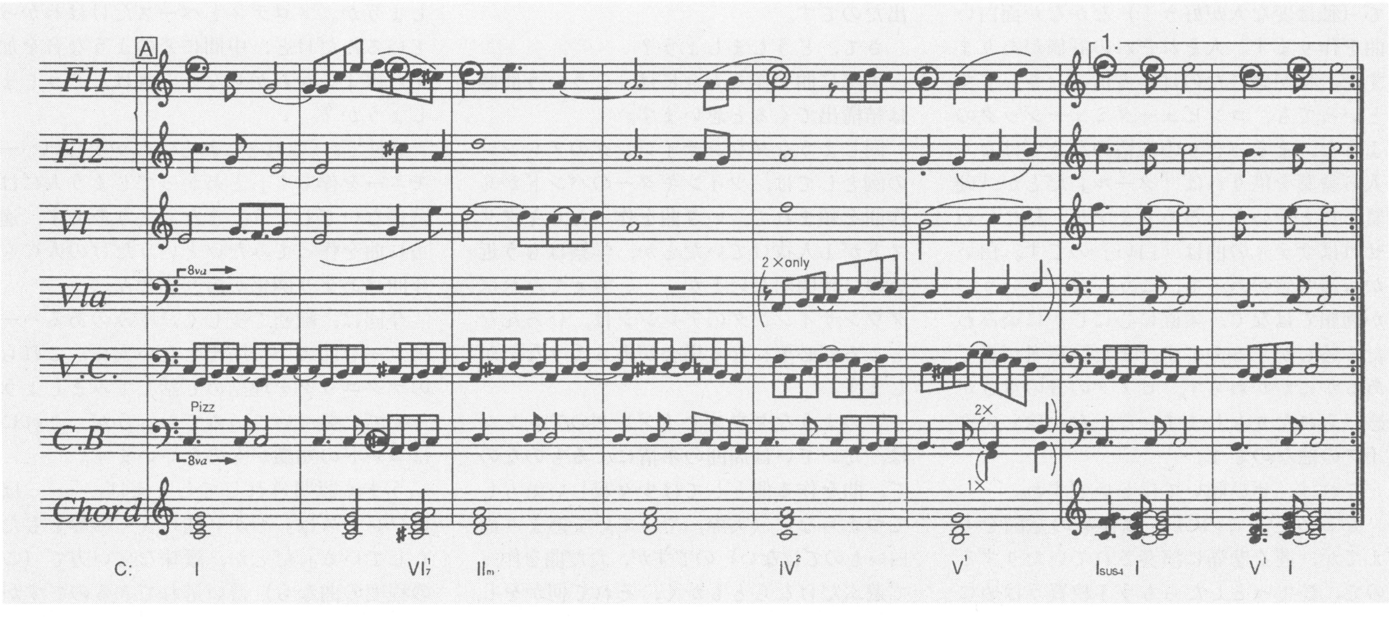



図4　省略

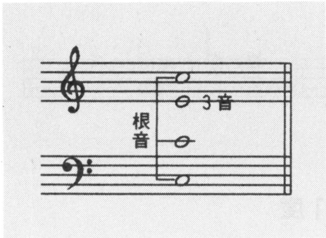



図5　各声部の間隔

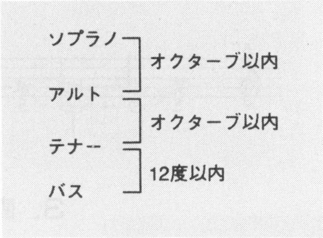


図6　音域に該当するコードCの構成音

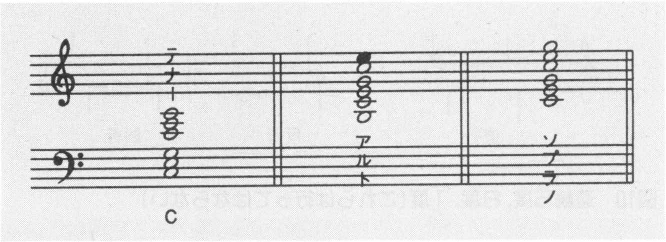


図7　密集配置

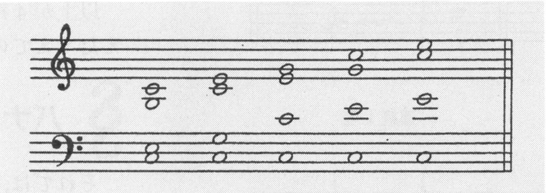
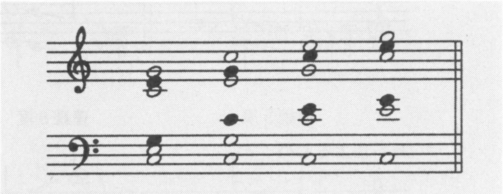



図8　開離配置



の3つを（アルト，テナー，バス）を下3声（か3せい）といいます。これらは意味そのままだから簡単にわかりますよね。

これらにはそれぞれ音域がありますので，それらを図2に記しておきます。まあ，これも，自分の声の音域からもある程度想像できますから，覚えるのもそれほど面倒なことではないでしょう。ちなみに図中の黒丸の音符は，発声が困難である，つまり例外的な発声域であまり使われない（ハーモニーに異常をきたさないためにも使うべきではない）部分です。

この4つの音に，コードに乗った音を配置していきます。ここでは話を単純化するために，トライアドコード（三和音）だけで話を進めていきます。

4声体は4つの音が必要なのに，コードは三和音です。ということは，必ず三和音の構成音のうち，何かが重複してしまうことになります。少し考えれば予想できるとおり，いちばん安定する良好な形は，根音を重複する形です。ただし，最も安定する形は根音重複ですが，3度，5度音が重複しても別にかまいません。ちなみに3度音の重複にはなにかと制限があるので，慣れないうちはあまり使わないほうがいいかもしれません（図3）。

根音が3音重複することが条件で，5度音なら省略することも可能ですが，これはお勧めできる形ではありません。やはり省略は特殊な形で，ハーモニーに穴を作りやすいからです。ところで，なぜ3度の音を省略してはいけないかはわかりますよね？　なぜなら，3度の音は特徴音だからです。

実は，楽譜1の「バナナパフェ味のそよ風」は3，4小節目のIImのところで3度の省略をしています。いわれてみれば3，4小節目はハーモニーに甘さがあるでしょう？　あとでもう少しお話ししますが，コードの特徴音がないのでメジャーなのかマイナーなのかもわかりません。練りこまれた曲とそうでもない曲の違いはここにあり，ってとこでしょうか。

ちなみに，根音は絶対に省略してはいけません（図4）。クォードコード以上においては，「根音省略形」という特殊な形があり，このときの4声体の配置では当然根音は省略されています。しかしそれは，根音を省いても3音以上のハーモニーがあって美しさが引き立つのであって，三和音でこれをやるとただのインターバルになってしまい，ハーモニーが興醒めなのです

話を先に進めましょう。

4声体の各音の配置には，それぞれの間隔に限界があります。

図5を見てください。上から順にソプラノとアルトの間はオクターブ（8度）以内，アルトとテナーの間もオクターブ以内，テナーとバスの間は12度以内です。曲が浮き足立ってしまうとか，重すぎてしようがないとかいうことは，いい加減に作っていくと，よくぶつかる問題なのですが，たいていの場合はこの条件を満たしていないことが原因です。たとえそうでなかったにしても，4声体の配置を見てみると，その原因がわかることがあります。原因がわかれば対処はできるので，4声体を置いてみる技を身につければ，便利なこときわまりないのです。

§　決められたバスの上の4声体の配置

当然のことながら，4声体の各声部の音域の条件を守り，配置の間隔の条件を守り，重複，省略などの項目を守ったとしても，同じコード，たとえばCなどで配置される音の場所は実にさまざまです。これらの配置は大きく分けると，次の2種類に分類されます。

・密集配置（クローズヴォイシング）
・開離配置（オープンヴォイシング）

これらの配置は，決められたバス（ここではCmajの根音C）の上に配置されるものです。

個々に見ていきましょう。

1．密集配置

文字どおり密集なのですが，音が直接隣接しているのではなく，コードの構成音どうしでの密集です。

図6を見てください。これは，上3声の個々の音域に該当しているコードCの構成音です。すなわち上3声の音は，各声部の限界を超えていなければ，これらの音から自由に音を選び出すことができます。

ここでいう密集とは，つまりそういうことで，コードの構成音が隣どうしであることを意味します。

さて，密集配置ですが，アルトの最も低い音Gから見ても，最も低めの音の配置は，下からテナーE，アルトG，ソプラノCになります。

このようにして，この3つすなわち上3声を必ず密集させて，できるかぎり上げていくと，コードCでの密集配置では，図7の5つになります。

2．開離配置

密集配置とは逆に，上3声を密着せずに，コードの構成音1音分を間に入れて，間隔をあけて配置します。

図6に示した音の配置を守って，これも同じように配置すると，コードCでの開離配置は図8にある4つになります。

図中の黒丸の音符は，間に1つあることを示しているものであって，実際の配置では存在しないものです。

ちなみに，これらの2つの配置は，来月と再来月の「ストリングスの使い方」「ブラスの使い方」（入れ替わるかもしれないけど）で密接にかかわり合ってきますから，

図9　3つの進行

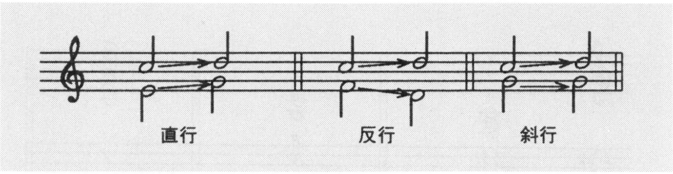


図11　これらはよい

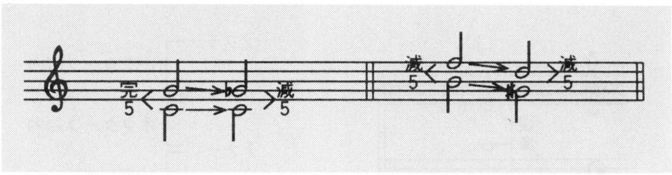


図10　連続5度，8度，1度(これらは行ってはならない)

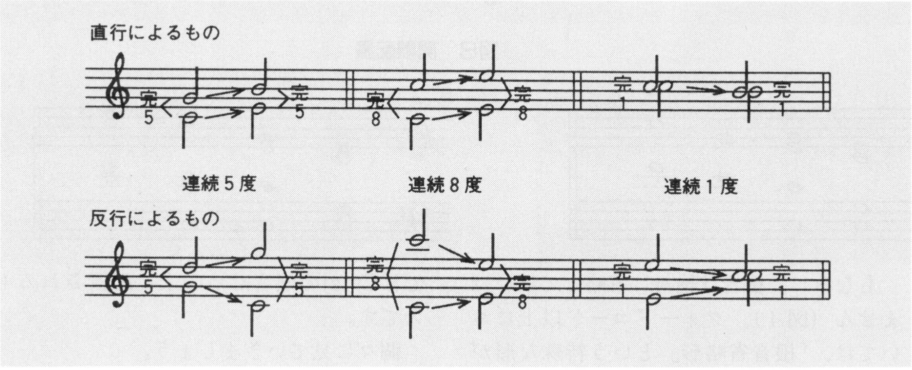
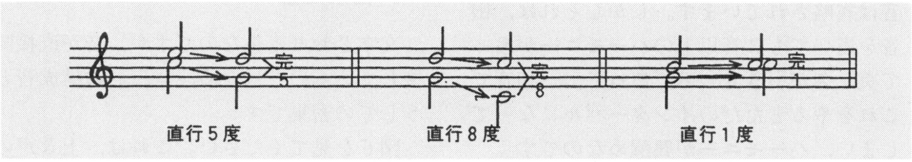


図12　直行5度，8(1)度



覚えておいてください。

§ 4声体での各音の進行

4声体をコード進行と照らし合わせてみると，コード進行（和声進行）というものは，4つの各声部の水平的なつながり，すなわち，各声部の音の変移といえます。

和声進行の2声部を考えたうえで，進行は次の3つに分類されます（図9）。

・直行　2声部が同じ方向へ進行
・反行　2声部が反対方向へ進行
・斜行　1声部のみが動き，もう片方は動かない

さて，この4声体の和声進行にも，やはり，やってはいけない進行，注意しなくてはならない進行というものがあり，それらは次の3つになります。

1．連続5，8(1)度

2声部が完全5，8(1)度の間隔を保って進行するものです。これらは直行によるものでも，反行によるものでも，やってはいけません。理由はハーモニーに単調さを招きやすいからです（図10）。

ただし，連続5度進行の場合は，あとのほうが減5度ならば進行してもかまいません（図11）。

勘違いしては困るのが，ユニゾンとは別だということです。ユニゾンとはメロディの動きのうえで完全8(1)度でハモりながら演奏するもので，ここで禁じているのは和声進行のことです。

2．直行5度，8(1)度

2声部が直行して，完全5度，完全8(1)度の形になることを，直行5度，8(1)度といいます(図12)。直行5度，8(1)度進行もハーモニーが軽くなりやすいので，あまり勧められる形ではありません。

このなかで，2声部が両方とも外声である場合，ハーモニーの単純化が目立ちやすいので，これは禁じられています。ただし，外声の場合でもソプラノが2度進行するものなら，経過的な音の推移といった理由から，この進行が許されます(図13)。

2声部の片方が内声であった場合は，常に許されます。理由は外声に比べて目立たないからです(図14)。

3．直行1度

すべてにおいて不可能です。

以上が4声体における，ハーモニーを考えたうえでの基礎知識です。

§ バナナを4声体にしてみる

それでは，この知識がどのようなところで役に立つかを知るためにも，問題の「バナナパフェ味のそよ風」を4声体に直してみましょう。

まず，最初の1小節のコードと，ベースノートに注目します。比較的ノーマルにトニックから始まる曲ですので，スケールCでコードはC(maj)，ベースノートはその根音のCです。

このバスは根音どおりのC，この音はコントラバス(C.B)で保たれています。

次はテナー，これはチェロ(V.C)の音が，コントラバスより少し上（といっても1オクターブほど）ということから，割り当てられるでしょう。

アルトに該当する音は，この曲のアンサ

図13　直行5度，8(1)度のよい例と悪い例

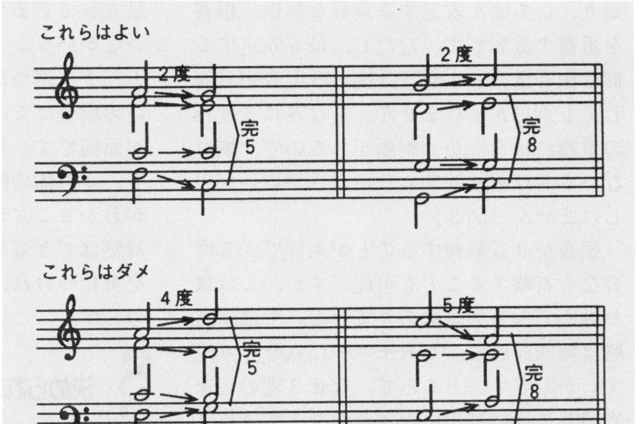
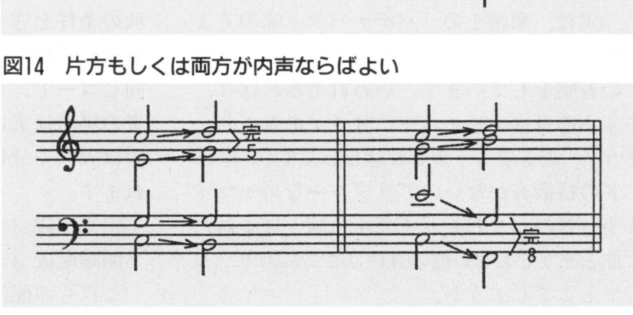


図14　片方もしくは両方が内声ならばよい

ンブルではヴィオラ(Vla)が最も近いのですが，この楽器はあとにしか出てきません。しかし，その代わりにイントロではヴァイオリンの音が低いところまで下りているので，これがアルトをとっているとみてよいでしょう。

ソプラノはいうまでもなく，フルート1(Fl1)です。

結局，1小節目の和声は，C，G，E，Cと，開離配置になっていることがわかります。これには，大きな理由がありますが，今月はそこまで進めないので，来月の課題にしておきましょう。

ソプラノの進行を見るために，楽譜1のフルート1のパートを見てください。音符を丸で囲んであるものが，ソプラノ進行を決定した音です。

ソプラノ進行がわかり，さらに開離配置とわかっているので，ここでは理想的な和声進行を作成するために，バナナの楽譜は見ないでやってみましょう（バナナは4声体を厳密に考えて作ったのではありません）。

まず，次の和音に共通の音があるかどうかをチェックします。2小節目ではコードはA7，すると前のテナーに共通の音がそのままGとしてあります。アルトの音も同様にEとして存在します。

しかし，この配置では，根音であるAが存在しなくなってしまいます。これではハーモニーに支障をきたします。また，ここのコードはクォードなので4声体にするにはすべての音を使用するか，たとえ省略したとしても，5度の音にしなければなりません。

都合の悪いことに，5度の音はソプラノで使われています。そうなると，アルトとテナーはA，Gのどちらかにしなくてはなりません。私はアルトとのバランスから，テナーにA，アルトにGを置くことにします。

3小節目でも同様に，共通の音からテナーはAを選択します。ここのアルトは共通音がありません。共通音がないときはなるべく近い音に進行させます。これだけの条件ではA，Fのどちらにでも進めますが，「開離配置」という条件が最初にあるので，Aに進むとソプラノのDと密着してしまうため，ここではFに進むのが妥当です。

5小節目ではテナーもアルトも，どちらも共通音がありますので，そのまま同じ音を保ちます。

このようにして配置していくと，バナナの理想的な和声進行は図15に示したようになります。

さて。

できた和声進行は，このバナナの曲で最も重要な音を表していることになるので，これらの音を踏むように作れば，厚みのあるしっかりとしたハーモニーで曲を作ることができます。

実際のバナナと照らし合わせてみます。

バナナは私の感性そのままで作られているので，一部分あまり芳しくない進行をしているかもしれません（無責任な……）。

まず3小節目，Fの音をどのパートも踏んでいません。このFは3度の音，すなわち，三和音では特徴音……。しかし，よくみると，DとAしかコードにのっていません。インターバルですね。これじゃあ，メジャーなのかマイナーなのかわからないし，ハーモニーに甘さを招きかねません。そこが穴だと思いながら聴き始めると，だんだんそこがものすごく薄く感じられたりして……。

まあ，ここはまだしも，7，8小節目などはアンサンブルがめちゃめちゃですね。MIDI楽器の厚めの音色で，ごまかしているという感じです。

というように，しっかり4声体に置き換えてみると，実は曲の構成の穴までわかってしまうというわけです。穴がわかればそれなりの修正ができますね。

§ 4声体の役に立つところ

和声進行を4声体で組むことのいちばんありがたいところは，まず，ハーモニーのボロが明確にわかる点です。

耳と感性だけで完全なハーモニーが作れる人はともかく，私のような凡人には，これは非常に嬉しいことです。場合によってはなんの音が腐ってるかなんてわからなくなることもありますし，なかなか耳だけでは正確で美しいハーモニーは作れませんからね。できる人は限られた人ですよ。

ま，そうでなかったとしても，メロディを作り，バス進行を決め，あとはいろいろな音を加えていく段階になったときに，楽譜の下のほうにこれを書いておくだけで，あとはどのパートにどんな音を加えたらよいかが，これまたハッキリわかります。それに，音がどのへんで薄くなっているかも一目瞭然ですから，パートを加えていく段階でこれほど便利なものはありません。

それから，耳がそれほど訓練されていなくても，理詰めで理想的（に近いといったほうが賢明？）なハーモニーがわかるので，パートを入れるときに「ここの音はどっちのほうがよいのかな？」という迷いも同時に解決されやすいのも，嬉しいところでしょう。

ほかにも恩恵はあります。

曲の和声進行のうえでの4声体がわかっていれば，アドリブ演奏もかなり楽に演奏できます。でもこれは，コンピュータミュージックしかやっていない人には，あまり関係ないかもしれませんね。

§ おわりに

……今回は，非常に苦しかった。先々月号でか，面白くないといわれてしまった手前，理詰めで云々は避けたかったんですけれど，このあとのことを考えると，これだけは説明せざるをえませんでした。

初めて4声体にとりかかったとき，非常に面倒臭かった思い出があります。実はまだまだ法則性はあるんですよね。

導音は主音にしか進めないとか，テナーにおける導音とバスにおけるV音が共に主音に進行するときの直行1度進行は許されるとか，苦労したなあ。

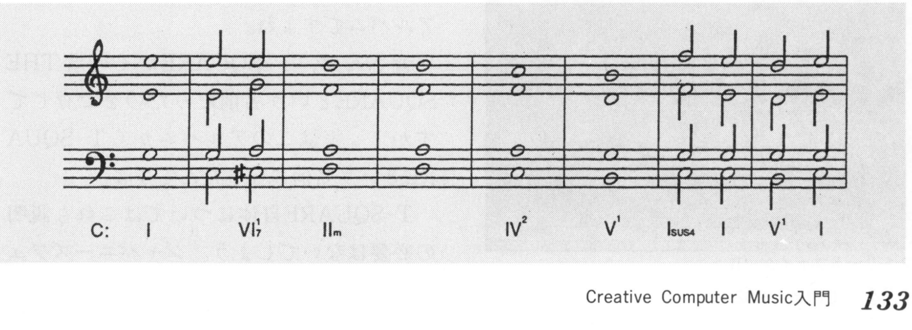
そんなわけで，それほど細かく4声体の規則を覚えなくても，ある程度骨組みが構築できれば，曲作りにもかなり貢献してくれます。

さて，お次はオーケストレーションのアレンジということで，まずはストリングスの利用法です。その次はブラスでもやりましょうか。

ばっちり覚えたら，もうフルオーケストラの曲もグ～さ！　なんてことはないんですけれど，役に立つこと請け合いでしょう。

それでは，また来月。

図15 「バナナ」の理想的な和声進行



Oh!X LIVE in '92

X68000・Z-MUSIC用 (SC-55対応) ©CAPCOM ストリートファイターIIより
BALROGのテーマ
Watanabe Kazuhiko 渡辺 一彦

X68000・Z-MUSIC+PCM8用
ARCADIA
Okada Kazuhiko 岡田 一彦

X1・musicBASIC用 ©Nintendo スーパーマリオブラザーズより
地上のテーマ
Nakamura Naoya 中村 直哉

ゲームミュージック2曲とT-SQUAREです。移植が待たれるストリートファイターIIは2月号でリュウのテーマを紹介しましたが，今回はBALROGのテーマです。他2曲も，Oh!Xの読者にはきっと馴染み深い曲なので，ぜひ聴いてみてくださいね。

ヒョロレイッヒ〜

さて，今月の1曲目は，いまだに人気絶好調のストリートファイターIIより，「BALROGのテーマ」をお届けしましょう。Z-MUSICシステム用でSC-55が必要です。

いまさらストリートファイターIIに関する説明はいらないでしょう。ゲームセンターを席巻し，スーパーファミコンをも巻き込み，本やCDまでが売れに売れているとあれば，あとはX68000用を待つのみといった感じですよね。たとえハードディスクの空きが12Mバイト必要だろうとメモリが4Mバイト必要だろうと専用ジョイスティックが必要だとしても，発売していただきたいものです。お願いね，カプコンさん。

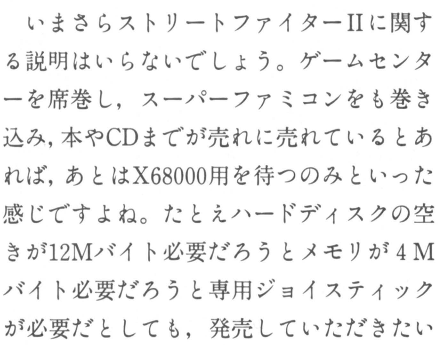
曲はBALROGがスペイン国籍ということもあって(？)，フラメンコをアレンジしたような曲になっています。この作品では若干のアレンジが加えられているようで，かっこよくまとまっています。

注意点として，トラックバッファが64Kバイトでは足りないため，標準のシステムではエラーが発生してしまいます。

ZMUSIC -T100

などとして，トラックバッファを70Kバイト以上確保しておいてください。

渡辺君はトラックバッファのアサインをリストの最後にとるという方法をとっていましたが，最新のZ-MUSICシステム(バージョン1.35)を使用した場合に演奏されないことが判明しましたので，アサインの部分をリストの上のほうに持ってくるという修正をさせていただきました。また，14トラック分しか演奏データがないにもかかわらず，無意味に16トラックを確保していたのも修正しておきました。もちろん曲データの変更はしていませんので，スピーカーから奏でられる作品には影響ありません。



ストリートファイターII

変身！ T-SQUARE

さて，2曲目は，同じくZ-MUSICシステム用にT-SQUAREの「ARCADIA」をお届けしましょう。演奏にはPCM8.Xが必要です。「ARCADIA」はアルバム「WAVE」からの選曲になります。「WAVE」といえば昔からのT-SQUAREファンには思い出深いアルバムですよね。

皆さんは，T-SQUAREがその昔THE SQUAREという名前だったのをご存じですか？　実はこのアルバムからT-SQUAREという名前になったんですよ。

T-SQUARE自体についてはこれも説明の必要はないでしょう。ジャパニーズフュージョンでは押しも押されもせぬトップスターですよね。F1のテーマとなっている「TRUTH」がやはり有名どころ。過去にもこのページを何度となく飾ってもらっています。

作品のほうは内蔵音源のみで再生されているだけあって，原曲と聴き比べるのは酷というものですが，FM音源で爽快なイメージの曲を爽快に演奏しています。MIDIを使えば表現は広がったかもしれませんが，掲載するには十分なレベルに達しています。特にリードの音色や使い方がうまく表現されています。

リストはちょっと長めですが，冬の夜長の指の運動にはもってこいですよね。苦労した分は作品のデキで元が取れると思いますので入力してみてください。

そういえば，今月のX68000用の曲の作者は2人とも一彦君なんですよね。ちょっと因縁めいてますが，これからもよきライバル(？)として頑張ってくださいね。

スーパーマリオBROS.

さて，X1には，MusicBASIC用にスーパーマリオブラザーズより「地上のテーマ」をお届けしましょう。X68000風の音色の拡張が必要です。PSGは使っていないので，ミキシングなどの心配は無用です。

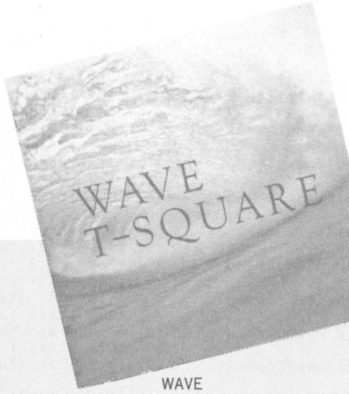
この曲は説明するまでもないでしょう。スーパーマリオブラザーズでスタートボタンを押して，最初の面の最初に流れるあの曲です。

そういえば大昔にX1turbo用のスーパーマリオブラザーズがあったのをご存じですか？　確かFM音源は未対応でPSGでBGMが鳴っていたというように記憶しています。実は私が編集室にあったX1turboIIでそのゲームをプレイしているときに電源が壊れてしまい，リセットはおろか電源を入れ直してもスーパーマリオが動きっぱなしだったことがありました。まさに電源ポンだったのでファミコンのようでしたが，背面のメインスイッチで電源を入れ直してもスーパーマリオだったときは，心底あせりました。

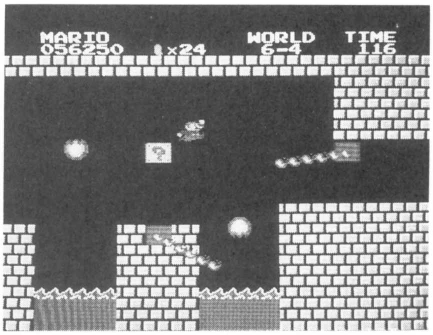
話がそれてしまいましたが，曲のデキは保証します。リストも短めですので，ぜひとも入力しましょう。さすがに最近はX1ユーザーのパワーが落ちてきてしまってますよね。入力するだけでなく投稿する元気も欲しいと思う今日この頃です。

作者の中村君は「ねこバス」のイメージが強く，X1の音楽ではかなり常連になりつつありました。ところが，最近X68000を買われたそうで，この作品がX1での最後の作品ということです。ちょっと残念ですが，いつの日かX68000用で復活してくれることでしょう(CM-300も買ったそうですし)。

(S. K.)



WAVE



スーパーマリオブラザーズ

リスト1　ストリートファイターII

```
     1: (i)
     2: (b1)
     3: .ROLAND_EXCLUSIVE $10,$42={
     4:                   $40,$00,$7F,$00}
     5: .SC55_V_RESERVE ={0,0,0,0,0,0,1,2,2,0,5,0,0,0,0,0}
     6:
     7: (m1,5000)
     8: (m2,5000)
     9: (m3,5000)
    10: (m4,5000)
    11: (m5,5000)
    12: (m6,5000)
    13: (m7,5000)
    14: (m8,5000)
    15: (m9,5000)
    16: (m10,5000)
    17: (m11,5000)
    18: (m12,5000)
    19: (m13,5000)
    20: (m14,5000)
    21:
    22: (a1,1)
    23: (a2,2)
    24: (a3,3)
    25: (a4,4)
    26: (a5,5)
    27: (a6,6)
    28: (a7,7)
    29: (a10,8)
    30: (a10,9)
    31: (a10,10)
    32: (a10,11)
    33: (a8,12)
    34: (a9,13)
    35: (a11,14)
    36:
    37: (o154)
    38: (t1) k0 r8 n1 @k3 @i$41,$10,$42 @m   0,  0,  0, 32 @u127
 @e80,32 @p64 @057 @v125 l16 [do] o4
    39: (t1) r1 r1 r1 r1
    40: (t1) e2<c2& c4>b4a6g6f6 edefe2. r4>b8<c8d6e6f6 e2<e2& e4
d8e8d6c6>b6 e1 r4d8e8f6g6g#6
    41: (t1) a2<c2& c6>b6<c6d6c6>b6 efede2.& e2r4e4 b<c>bga2.& a
6b6<c6c#6d6d#6 efed#e2.& e2r2>
    42: (t1) |:2 a4<c4&c8>b<cdc>ba b8.ee4rb<cdedc>b< c8.>e<e2&e8
de d2..r8 > | r4f4<d4.c>b<crr>e e2.r4 f#4<d6c6d6 >b<c>bab2r4 :|
< cdefgfede2 < cdefgfede2 [loop]
    43: (t2) k0 r8 n2 r16 @k-3 @i$41,$10,$42 @m  0,  0,  0, 32 @
u127 @e80,32 @p32 @057 @v122 l16 [do] o4
    44: (t2) r1 r1 r1 r1
    45: (t2) e2<c2& c4>b4a6g6f6 edefe2. r4>b8<c8d6e6f6 e2<e2& e4
d8e8d6c6>b6 e1 r4d8e8f6g6g#6
    46: (t2) a2<c2& c6>b6<c6d6c6>b6 efede2.& e2r4e4 b<c>bga2.& a
6b6<c6c#6d6d#6 efed#e2.& e2r2>
    47: (t2) |:2 a4<c4&c8>b<cdc>ba b8.ee4rb<cdedc>b< c8.>e<e2&e8
de d2..r8 > | r4f4<d4.c>b<crr>e e1 f#4<d6c6d6 >b<c>bab2. :| < cd
efgfede4.. cdefgfede2 r [loop]
    48: (t3) k0 @k4 @l22 r n3 @i$41,$10,$42 i0 @e72,24 @49 @p70
o5 l16 @u127 @v114 [do]
    49: (t3) |:2 |:2 'ec>a<' r 'ec>a<' 'ec>a<' :| r |:4 'ec>a<'
:| r | 'fd>b<' r |:2 'fd>b<' r 'fd>b<' 'fd>b<' :| r |:3 'fd>b< '
 :| 'gec' r 'fd>b<' r :| 'd>bg<' r |:2 'd>bg' r 'd>bg<' 'd>bg<'
:| r |:4 'd>bg<' :| r 'd>bg' r
    50: (t3) |:2 |:2 'ec>a<' r 'ec>a<' 'ec>a<' :| r |:4 'ec>a<'
:| r | 'fd>b<' r |:2 'fd>b<' r 'fd>b<' 'fd>b<' :| r |:3 'fd>b< '
 :| 'gec' r 'fd>b<' r :| 'd>bg<' r |:2 'd>bg' r 'd>bg<' 'd>bg<'
:| r |:4 'd>bg<' :| r 'd>bg' r
    51: (t3) |:2 |:2 'ec>a<' r 'ec>a<' 'ec>a<' :| r |:4 'ec>a<'
:| r | 'fd>b<' r |:2 'fd>b<' r 'fd>b<' 'fd>b<' :| r |:3 'fd>b< '
 :| 'gec' r 'fd>b<' r :| 'd>bg<' r |:2 'd>bg' r 'd>bg<' 'd>bg<'
:| r |:4 'd>bg<' :| r 'd>bg' r
    52: (t3) |:2 |:2 'fc>a<' r 'fc>a<' 'fc>a<' :| r |:3 'fc>a<'
:| | 'fc>a< ' r 'fc>a<'r :| 'gd>b<' r 'fc>a<' r |:2 |:2 'e>bg#<'
 r 'e>bg#<' 'e>bg#' :| r |:4 'e>bg#<' :| r 'e>bg#<' r :|
    53: (t3) |:2 'fc>a<' r 'fc>a<' 'fc>a<' :| r |:4 'fc>a<' :| r
 'fc>a<' r |:2 'gd>b<' r 'gd>b,' 'gd>b<' :| r |:4 'gd>b<' :| r '
gd>b<' r 'a*384ec#'
    54: (t3) 'ec>a<' rrr 'ec>a<' r 'ec>a<' 'ec>a<' r |:4 'ec>a<'
 :| r 'ec>a<' r |:2 'd>bg#<' r 'd>bg#<' 'd>bg#<' :| r |:4 'd>bg#
>' :| r 'ec>a<' r
    55: (t3) |:2 'ec>a<' r 'ec>a<' 'ec>a<' :| r |:4 'ec>a<' :| r
 'afd' r |:2 'afd' r 'afd' 'afd' :| r |:4 'afd' :| r 'fd>a<' r
    56: (t3) |:2 'fd>a<' r 'fd>a<' 'fd>a<' :| r |:4 'fd>a<' :| r
 'ec>a<' r |:2 'ec>a<' r 'ec>a<' 'ec>a<' :| r |:4 'ec>a<' :| r '
d#>bf#<' r
    57: (t3) |:2 'd#>bf#<' r 'd#>bf#' 'd#>bf#<' :| r |:4 'd#>bf#
<' :| r 'aec' r |:2 'aec' r 'aec' 'aec' :| r |:4 'g#ec' :| r 'g#
ec' r
    58: (t3) 'ec>a<' rrr 'ec>a<' r 'ec>a<' 'ec>a<' r |:4 'ec>a<'
 :| r 'ec>a<' r |:2 'd>bg#<' r 'd>bg#<' 'd>bg#<' :| r |:4 'd>bg#
>' :| r 'ec>a<' r
    59: (t3) |:2 'ec>a<' r 'ec>a<' 'ec>a<' :| r |:4 'ec>a<' :| r
 'afd' r |:2 'afd' r 'afd' 'afd' :| r |:4 'afd' :| r 'afd<' r
    60: (t3) 'e>bg#<' r > 'e>bg#<' 'e>bg#<' < 'e2>bg#<' 'e4>bg#<
'  'g#e>b<' r > 'g#e>b<' 'g#e>b<d' < 'g#2e>b<' 'g#4e>b<' [loop]
    61: (t7) k0 r8 n7 @i$41,$10,$42 @e96,32 @u127 @38 @p44 l16 @
v125 [do] o1
    62: (t7) |:11 a8<a8>a8<a>ara<a8>a8<a8> :| < e8<e8>e8<e>ere<e
8>e8<e8> >
    63: (t7) |:2 f8<f8>f8<f>frf<f8>f8<f8> | d8<d8>d8<d>drd<d8>d8
<d8> |:2 e8<e8>e8<e>ere<e8>e8<e8> :| :| g8<g8>g8<g>grg<g8>g8<g8
aaaaa2. a4a4aa>a8<aa>a<a >
    64: (t7) a8<aa>a8<a>ara<aa>a8<a8> g#8<g#g#>g#8<g#>g#rg#<g#g#
>g#8<g#8> g8<gg>g8<g>grg<gg>g8<g8> f#8<f#f#>f#8<f#>f#rf#<f#f#>f#
8<f#8>
    65: (t7) < d8<dd>d8<d>drd<dd>d8<d8> e8<ee>e8<e>ere<ee>e8<e8>
 d#8<d#d#>d#8<d#>d#rd#<d#d#>d#8<d#8> e8<ee>e8<e>ere<ee>defd >
    66: (t7) a8<aa>a8<a>ara<aa>a8<a8> g#8<g#g#>g#8<g#>g#rg#<g#g#
>g#8<g#8> g8<gg>g8<g>grg<gg>g8<g8> f#8<f#f#>f#8<f#>f#rf#<f#f#>f#
8<f#8>
    67: (t7) < e2.<ed>be > e2.<<ee8. [loop]
    68: (t8) r8 l16 o2 i17 @v96 [do]
    69: (t8) |:2 ccrrd8.c rccrd4 cc8.d8.c rccc | d8g8 :| dcdd
    70: (t8) |:2 ccrrd8.c rccrd8.c cccrd8.c rccc | dbgr :| drdd
    71: (t8) |:2 ccrrd8.c rccrd8.c cccrd8.c rccc | dbgr :| d<d>b
g
    72: (t8) |:4 ccrrd8.c | rccr d8g8 :| r<d>bg d32d32dcd
    73: (t8) |:2 ccrrd8.c rccr | d8g8 :| d<d>bg c4c4c4c4 <d32d32
ddddd>b8 <d32d32ddd>bbgg
    74: (t8) |:2 crccd8.c rccrd8b8 ccrrd8.c rcccd8g8  crccd8.c r
cccd8b8 ccccd8.c | rrcc drdd :| <d32d32ddd> b32b32bbb
    75: (t8) crccd8.c rccrd8b8 ccrrd8.c rcccd8g8  crccd8.c rcccd
8b8 ccccd8.c rrcc drdd c4c4c4d<d>bg c4c4c4d32d32ddd [loop]
    76: (t9) r8 @v96 l16 o2
    77: (t9) [do] |:18 |:14 f# :| a#8 :| |:8 f#4 :|  |:12 |:3 f#
rf#f# :| f#f#a#8 :| |:8 f#4 :| [loop]
    78: (t10) r8 @v96 o3
    79: (t10) [do] c#1 r1 c#1 r2.a4
    80: (t10) |:4 c#1 r1 r1 | r2.a4 :| r1
    81: (t10) |: c#1 r1 r1 r1 | c#1 r1 r1 r2.a4 :| c#1 a1 [loop]
    82: (t11) @v96 r8 l8
    83: (t11) o5 r16..b [do] |:71 rb  :| r1 r1 |:48 rb :| r1r1 r
b [loop]
    84: (t12) k0 n8 @k3 r8 @i$41,$10,$42 @e64,28 @u127 i1 @100 o
```

```
3 l16 @v112
(t12) [do] r1 r1 r1 r2 @p000b<@p018c@p036e@p054a@p072b<@
p090c@p108e@p127a>>  r1 r1 r1 r1
(t12) |:2 r1 r1 r1 r2 @p000b<@p018c@p036e@p054a@p072b<@p
090c@p108e@p127a>> :|
(t12) r1 r1 r1 @p34 << aarrarrr >> @p000b<@p018c@p036e@p
054a@p072b<@p090c@p108e@p127a>>
(t12) r1 r1 |:2 r1 r1 r1 r1 r1 r2 @p000b<@p018c@p036e@p0
54a@p072b<@p090c@p108e@p127a>> :| [loop]
(t13) k0 n9 @k-3 r8 @i$41,$10,$42 @e64,28 @u127 i1 @100
o3 l16 @v108
(t13) r [do] r1 r1 r1 r2 @p000b<@p018c@p036e@p054a@p072b
<@p090c@p108e@p127a>>  r1 r1 r1 r1
(t13) |:2 r1 r1 r1 r2 @p000b<@p018c@p036e@p054a@p072b<@p
090c@p108e@p127a>> :|
(t13) r1 r1 r2... @p94 ¯4 <<< eerrerrr >>> _4 r @p000b<@
p018c@p036e@p054a@p072b<@p090c@p108e@p127a>>
(t13) r1 r1 |:2 r1 r1 r1 r1 r1 r2 @p000b<@p018c@p036e@p0
54a@p072b<@p090c@p108e@p127a>> :| [loop]
(t14) k0 @k-4 @l22 r n11 @i$41,$10,$42 @e72,32 i0 @26 @p
54 o4 l16 @u127 @v116 [do]
(t14) |:2 |:3 'e8c>a<' 'ec>a<' 'ec>a<' :| 'e8c>a<' | 'f8
c>a<' |:3 'f8c>a<' 'fc>a<' 'fc>a<' :| 'g8c>a<' 'f8c>a<' :| 'd8c>
a<' |:3 'd8>bg<' 'd>bg<' 'd>bg<' :| 'd8>bg<' 'e8>bg<'
(t14) |:2 |:3 'e8c>a<' 'ec>a<' 'ec>a<' :| 'e8c>a<' | 'f8
c>a<' |:3 'f8c>a<' 'fc>a<' 'fc>a<' :| 'g8c>a<' 'f8c>a<' :| 'd8c>
a<' |:3 'd8>bg<' 'd>bg<' 'd>bg<' :| 'd8>bg<' 'e8>bg<'
(t14) |:2 |:3 'e8c>a<' 'ec>a<' 'ec>a<' :| 'e8c>a<' | 'f8
c>a<' |:3 'f8c>a<' 'fc>a<' 'fc>a<' :| 'g8c>a<' 'f8c>a<' :| 'd8c>
a<' |:3 'd8>bg<' 'd>bg<' 'd>bg<' :| 'd8>bg<' 'e8>bg<'
(t14) |:2 |:3 'f8c>a<' 'fc>a<' 'fc>a<' :| | |:2 'f8c>a<'
 :| :| 'g8c>a<' 'f8c>a<' |:2 |:3 'e8>bg#<' 'e>bg#<' 'e>bg#<' :|
|:2 'e8>bg#<' :| :|
(t14) |:3 'f8c>a<' 'fc>a<' 'fc>a<' :| |:2 'f8c>a<' :|  |
:3 'g8d>b<' 'gd>b<' 'gd>b<' :| |:2 'g8d>b<' :| |:4 'aec#' :| 'a2
.ec#' |:2 'a4ec#' :| |:8 'aec#' :|
(t14) 'e4c>a<' |:2 'e8c>a<' 'ec>a<' 'ec>a<' :| 'e8c>a<'
'd8c>a<'  |:3 'd8>bg#<' 'd>bg#<' 'd>bg#<' :| 'd8>bg#<' 'e8>bg#<'
(t14) |:3 'e8c>a<' 'ec>a<' 'ec>a<' :| 'e8c>a<' 'a8c>a<'
 |:3 'a8fd' 'afd' 'afd' :| 'a8fd' 'f8d>a<'
(t14) |:3 'f8d>a<' 'fd>a<' 'fd>a<' :| 'f8d>a<' 'e8d>a<'
 |:3 'e8c>a<' 'ec>a<' 'ec>a<' :| 'e8c>a<' 'd#8c>a<'
(t14) |:3 'd#8>bf#<' 'd#>bf#' 'd#>bf#<' :| 'd#8>bf#<' 'a
8>bf#<' |:2 'a8ec' 'aec' 'aec' :||:2 'g#8e>b<' 'g#e>b<' 'g#e>b<'
 :|
(t14) 'e4c>a<' |:2 'e8c>a<' 'ec>a<' 'ec>a<' :| 'e8c>a<'
'd8c>a<'  |:3 'd8>bg#<' 'd>bg#<' 'd>bg#<' :| 'd8>bg#<' 'e8>bg#<'
(t14) |:3 'e8c>a<' 'ec>a<' 'ec>a<' :| 'e8c>a<' 'a8c>a<'
 |:3 'a8fd' 'afd' 'afd' :| |:2 'a8fd' :|
(t14) 'e8>bg#<' > 'e>bg#<' 'e>bg#<' < 'e2>bg#<' 'e4>bg#<
'  'g#8e>b<' > 'g#e>b<' 'g#e>b<d' < 'g#2e>b<' 'g#4e>b<' [loop]
(p)
```

リスト2　ストリートファイターII用カウンタ表示

```
 1:00000018 00001980   2:00000024 00001980   3:00000016 00001980   7:00000018 00001980
 8:00000018 00001980   9:00000018 00001980  10:00000018 00001980  11:00000045 00001980
12:00000018 00001980  13:00000024 00001980  14:00000016 00001980
```

リスト3　ARCADIA

日本音楽著作権協会(出)許諾第9271590-201号

```
.comment ~ARCADIA~                    T-SQUARE      Pr
gramed by 岡田　一彦
/  for ZMUSIC.X & PCM8.X
/          PCM8.xを使用しています。
/          ＰＣＭバッファは、
/          １２０ＫＢ以上確保して下さい。
(i)
(b0)
(m1,2000)(aFm1,1)
(m2,8000)(aFm2,2)
(m3,8000)(aFm3,3)
(m4,4000)(aFm4,4)
(m5,4000)(aFm5,5)
(m6,2000)(aFm6,6)
(m7,2000)(aFm7,7)
(m8,2000)(aFm8,8)
(m9,2000)(aAdpcm,9)
(m10,2000)(aAdpcm,10)
(m11,2000)(aAdpcm,11)
(m12,2000)(aAdpcm,12)
(m13,2000)(aAdpcm,13)
(m14,2000)(aAdpcm,14)
(m15,2000)(aAdpcm,15)
(m16,2000)(aAdpcm,16)
.adpcm_block_data=arcadia.ZPD
/      AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  BASS
(@70, 31, 16,  0,  3,  2, 33,  0, 10,  0,  0,  0,
      31, 12,  0,  3,  2, 45,  0,  0,  0,  0,  0,
      31,  8,  0,  3,  2, 18,  1,  0,  0,  0,  0,
      31,  8,  0,  4,  2,  2,  1,  0,  0,  0,  0,
/      AL  FB  OM
        0,  7, 15)
/      AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Glass
(@71, 28, 12,  0,  4, 14, 30,  0, 13,  3,  0,  0,
      31, 11,  0,  7,  5,  4,  0,  1,  3,  0,  0,
      31,  7,  0,  4,  4, 30,  0, 12,  7,  0,  0,
      20, 11,  0,  7,  5,  8,  0,  1,  7,  0,  0,
/      AL  FB  OM
        4,  4, 15)
/      AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  D.G.
(@72, 16, 15,  0,  3,  3, 26,  0,  3,  3,  0,  0,
      19, 31,  0,  5,  0, 27,  0,  1,  0,  0,  0,
      26, 31,  0,  3,  0, 26,  0,  1,  0,  0,  0,
      27, 31,  0, 11,  0,  2,  0,  1,  0,  0,  0,
/      AL  FB  OM
        0,  7, 15)
/      AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Str.
(@73, 31,  0,  0,  0,  0, 18,  0,  4,  3,  0,  0,
      17,  9,  0,  7,  1,  0,  0,  4,  3,  0,  0,
      31,  0,  0,  0,  0, 23,  0,  4,  7,  0,  0,
      17,  9,  0,  7,  1,  0,  0,  4,  7,  0,  0,
/      AL  FB  OM
        4,  4, 15)
/          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN  Piano
(v74,0,    60, 15,  2,  1,194, 24,  0,  5,  0,  3,  0
/      AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME
      31,  6,  0,  4,  3, 32,  0,  8,  7,  0,  0
      20, 10,  0,  6,  7,  4,  0,  8,  7,  0,  0
      31,  6,  0,  4,  3, 36,  0, 12,  3,  0,  0
      31, 11,  0,  6,  7,  4,  0,  8,  3,  0,  0)
/      AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Lyrico
n
(@75, 13,  9,  0,  8,  1, 26,  2,  1,  0,  0,  0,
      15,  8,  0,  8,  1, 21,  1,  2,  1,  0,  0,
      21, 14,  0,  8,  1,  4,  0,  1,  0,  0,  0,
      24,  3,  0,  8,  2,  1,  1,  1,  4,  0,  0,
/      AL  FB  OM
        5,  7, 15)
/      AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Gt(Bac
k)
(@76, 16, 10,  0,  5,  3, 26,  0,  1,  3,  0,  0,
      19,  4,  0,  4,  2, 21,  0,  1,  0,  0,  0,
      26,  4,  0,  4,  2, 24,  0,  0,  0,  0,  0,
      27, 12,  0,  6,  1,  3,  0,  1,  0,  0,  0,
/      AL  FB  OM
        0,  4, 15)
/      AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME  Gt(sol
o)
(@77, 16, 16,  0,  6,  3, 30,  0,  3,  3,  0,  0,
      19,  0,  0,  6,  0, 33,  0,  1,  0,  0,  0,
      26,  0,  0,  6,  0, 13,  0,  1,  0,  0,  0,
      27,  0,  0,  8,  0,  0,  0,  1,  0,  0,  0,
/      AL  FB  OM
        0,  7, 15)
/------------------------------------------------------------
/ MML DATA
/        Bass
/    I n t r o
(t1) t158@70l8o3q8v12
(t1) |:7r1:| r4.c2>e <e4r2. r2.f+d ec>a<d>bgf+e& eef+4g4
.e
(t1) |:<eee>eeeb<e d4.|c4.d4:| c2>e<eee>eeeb<e d4.c4.f+d
(t1) ec>a<d>bgf+e& eef+4g4.a& aab<c&c2& c2&cv14(c4,>c)v1
2e&
/   A  p a r t
(t1) |:|:eee4r4.f+& f+f+f+4r4.g& ggg4r4.a&
(t1) a<ec+>a<|d>bge&:||o3cd4>e&:|o3c4d4
/   B  p a r t
(t1) |:16e:| |:16d:| |:16c:| >bbb<c&cccc cccd&dddd>
/   C  p a r t
(t1) |:eeeeeb<ed& dddddd>a& aaaaaaab& bbbbbf+b|e&:|a&
```

▶DTM特集も回を重ねてきて，10月号はすいぶんと落ち着いた，実践的なものに仕上がっていますね。色モノよりもこれから少し深く突っ込んでいくための手助け，といった意味合いが強く好感が持てました。でも，ひとつぐらいはうけ狙いの記事がほしかったな。
佐藤　仁(23)静岡県

```
115: (t1) aaaaa<ead& ddddd>f+ag& ggggg<d>g<c& cccccc>g<c&
116: (t1) ccccccc>b& bbbbbbbf+ b4.<c4.d4 c4.d4.d+e>
117: /   D  part
118: (t1) e1& e1 e4r2. r4.<c2>e <e4r2. r2.f+d ec>a<d>bgf+e&
119: (t1) eef+4g4.e |:<eee>eeeb<e d4.|c4.d4:| c2>e <eee>eeeb<
e
120: (t1) d4.c4.f+d ec>a<d>bgf+e& eef+4g4.a& aab<c&ccde&
121: (t1) eef+g{raf+dge}2 {c>ab}4&b2r<c&
122: /   E  part
123: (t1) |:15c:|>b& bbbbbbbb& bbbbbbba& |:15a:|g& ggggggggf&
124: (t1) fffffffe |:13e:|b<ed& |:15d:|c& |:15c:|>b&
125: (t1) bb<b>bbb<b>b <b>b<b>b<b>bge&
126: /   F  part     (Apartと同一です)
127: (t1) |:|:eee4r4.f+& f+f+f+4r4.g& ggg4r4.a&
128: (t1) a<ec+>a<|d>bge&:||o3cd4>e&:|o3c4d4
129: /   B  part
130: (t1) |:16e:| |:16d:| |:16c:| >bbb<c&cccc cccd&dddd>
131: /   C  part(繰り返し記号が追加されています)
132: (t1) |:|:eeeeeb<ed& dddddd>a& aaaaaaab& bbbbbf+b|e&:|a&
133: (t1) aaaaa<ead& ddddd>f+ag& ggggg<d>g<c& cccccc>g<c&
134: (t1) ccccccc>b& bbbbbbbf+ b4.<c4.d4 c4.d4.d+e>:|
135: /   G  part
136: (t1) |:o2|:<eee>eeeb<e d4.|c4.d4:|c2>e
137: (t1) <eee>eeeb<e d4.c4.f+d ec>a<d>bgf+e& eef+4g4.e:|
138: (t1) o2|:<eee>eeeb<e |d4.c4.d4:|¥8d4.c2>e
139: (t1) <eee>eeeb<e d4.c4.f+d ec>a<d>bgf+e& eef+4g4.e
140:
141: /       Melody
142:
143: /   Intro
144: (t2) @75116@k194q8v14p3
145: (t2) |:8r1:|r2
146: (t2) @s6@m20@h44
147: (t2) o3_12|:c+ef+:|¯3|:ef+g+:|¯|:f+g+b:|¯a<c+d+>¯c+ef+
148: (t2) o4c+ef+ef+g+ef+g+ef+g+b<c+ec+ {f+ec+ef+}4{g+c+ef+g}
4b8<c+4.&
149: (t2) c+1&c+2&c+8.>bgf+ec+ f+c+ef+ge{f+g+f+}8ec+>bg+gf+ec
+
150: (t2) f+8ef+gef+4.r8f+8 g+8b8<c+8e8(e32,f+)&f+8..g+4
151: (t2) (e32,f+)&f+8..(d32,e)&e8..c+4>b8(g32,f+)&f+.&
152: (t2) f+8e8c+2. r1r1r1
153: /   トラック2と似ています(5行目以降同じ)
154: (t3) r96r8  |:3r1:|@7218v9o3@s3@m6@h24p2@k8r2..e
155: (t3) |:b<egb<cd>bg af+d|gecf+d>:|o4g&g4.>eb<egb<
156: (t3) @75116@k188q8v12p1
157: (t3) @s6@m20@h44
158: (t3) o3_12|:c+ef+:|¯3|:ef+g+:|¯|:f+g+b:|¯a<c+d+>¯c+ef+
159: (t3) o4c+ef+ef+g+ef+g+ef+g+b<c+ec+ {f+ec+ef+}4{g+c+ef+g}
4b8<c+4.&
160: (t3) c+1&c+2&c+8.>bgf+ec+ f+c+ef+ge{f+g+f+}8ec+>bg+gf+ec
+
161: (t3) f+8ef+gef+4.r8f+8 g+8b8<c+8e8(e32,f+)&f+8..g+4
162: (t3) (e32,f+)&f+8..(d32,e)&e8..c+4>b8(g32,f+)&f+.&
163: (t3) f+8e8c+2. r1r1r1
164: /   A  part(以下、Cまで同じ)
165: (t2) l8o4@h44_3
166: (t2) |:|:r4.(f+32,g+)&g+16.ec+e(e32,f+)&f+16.&
167: (t2) f+d+4>b&bg+b<|(e32,f+)&f+16.&
168: (t2) f+ee4.>b<e(f+32,g+)&g+16.& g+f+c+c+&c+2:|o4e&
169: (t2) ed+ef+g+ef+(g+32,a+)&a+16.& a+g+|
170: (t2) f+c+e(e32,f+)&f+8..r:|a+b¯3<c+>b<c+d+
171:
172: (t3) l8o4@h44_3
173: (t3) |:|:r4.(f+32,g+)&g+16.ec+e(e32,f+)&f+16.&
174: (t3) f+d+4>b&bg+b<|(e32,f+)&f+16.&
175: (t3) f+ee4.>b<e(f+32,g+)&g+16.& g+f+c+c+&c+2:|o4e&
176: (t3) ed+ef+g+ef+(g+32,a+)&a+16.& a+g+|
177: (t3) f+c+e(e32,f+)&f+8..r:|a+b¯3<c+>b<c+d+
178: /   B  part
179: (t2) (f+32,g+)&g+*138r4 (f+32,g+)&g+*66d+4.>b&<{c+&d+}
180: (t2) e2.r4 (d32,e)&e..&ec+4.>a4< (c+32,d+)&d+*138r4
181: (t2) (c+32,d+)&d+*66>b4.f+4 b4.<c+&c+2 c+4.d+4ef+g+&
182: (t3) (f+32,g+)&g+*138r4 (f+32,g+)&g+*66d+4.>b&<{c+&d+}
183: (t3) e2.r4 (d32,e)&e..&ec+4.>a4< (c+32,d+)&d+*138r4
184: (t3) (c+32,d+)&d+*66>b4.f+4 b4.<c+&c+2 c+4.d+4ef+g+&
185: /   C  part
186: (t2) @q1|:g+f+4e4d+4(e32,f+)&f+16.& f+e4d+4c+>b<c+&
187: (t2) c+2&c+c+|g+(e32,f+)&f+16.& f+2ref+(f+32,g+)&g+16.&:
|
188: (t2) f+(c+32,d+)&d+16.& d+2red+c+& c+4d+er(f+32,g+)&g+16
.rf+&
189: (t2) f+4.(f+32,g+)&g+16.&g+4r>b& b4<c+d+rf+r(d+32,e)&e16
.&
190: (t2) e2.r>a& a4b<c+r(d+32,e)&e16.rd+& d+q8c+4>b4ag+f+
191: (t2) g+f+g+abg+ab <c+>ab<c+d+ef+(f+32,g+)&g+16.&
192:
193: (t3) @q1|:g+f+4e4d+4(e32,f+)&f+16.& f+e4d+4c+>b<c+&
194: (t3) c+2&c+c+|g+(e32,f+)&f+16.& f+2ref+(f+32,g+)&g+16.&:
|
195: (t3) f+(c+32,d+)&d+16.& d+2red+c+& c+4d+er(f+32,g+)&g+16
.rf+&
196: (t3) f+4.(f+32,g+)&g+16.&g+4r>b& b4<c+d+rf+r(d+32,e)&e16
.&
197: (t3) e2.r>a& a4b<c+r(d+32,e)&e16.rd+& d+q8c+4>b4ag+f+
198: (t3) g+f+g+abg+ab <c+>ab<c+d+ef+(f+32,g+)&g+16.&
199: /   D  part(トラック2と3全く違う)
200: (t2) g+2..b <c+2.(f+32,g+)&g+..& g+1&
201: (t2) g+2_3r2 |:10r1:|
202: (t2) @k0@7218v11@s3@m6@h24o3
203: (t2) r2..e& eef+4g4.a& aab<c&ccde& eef+gr2 r1 v10p3
204: (t3) g+2..b <c+2.(f+32,g+)&g+..& g+1&
205: (t3) g+4@7218v9@k8p2@s3@m6@h24o4g&g4.r>e
206: (t3) |:b<egb<p3cd>bg af+dgecf+d ec>a<d>bgf+b& bb<d4d4.|>
e
207: (t3) |:b<egb<cd>bg af+d|gecf+d>:|o4c&c4.>e:|o4e&
208: (t3) eegg&ggab&bb<dd{r>af+dge}2 {c>ab}4&b4&(b4.,>>b)r v8
p3
209: /   E  part
210: (t2) @77o3l16q8g¯3ab¯al24b<cdf+gab&b<ddddeeeeee
211: (t2) |:6f+:|ggggggaaaaaal16ga(a64,b)&b..&
212: (t2) b4&(b12,a)g12f+12agf+egf+ed edc+>b<dc+>babageged>b
213: (t2) a8ga_3e2¯3g4& (g4,>g)&(g4,<g)g8g8g8g8&
214: (t2) g8<(a64,b)&b..&b8&(b,a)&(a,g)a4&(a8,>a)(a32,<a)&a.
215: (t2) bb+bb+bage-de-dcdc>ba gaf+ge4.<g8g4&
216: (t2) (g4,>g)&(g4,<g)g8g8g8>(e32,f+)&f+.& f+8<f+8f+4f+4
217: (t2) (c32,f+)&f+.a8& {af+<de}2(e32,f+)&f+.&e&de4&
218: (t2) (e4,>e)<l24|:4bg&af+d&e:|l16b+(a,g)&g&a(a32,b)&b.&
b4(a32,b)&b.&
219: (t2) (b4,a+)&a+8b8&(b4,>b)<r4 f+gf+gf+ed+dc+c-c+>bar8.
220: /   7行目以外は、トラック2と全く同じ
221: (t3) @77o3l16q8g¯3ab¯al24b<cdf+gab&b<ddddeeeeee
222: (t3) |:6f+:|ggggggaaaaaal16ga(a64,b)&b..&
223: (t3) b4&(b12,a)g12f+12agf+egf+ed edc+>b<dc+>babageged>b
224: (t3) a8ga_3e2¯3g4& (g4,>g)&(g4,<g)g8g8g8g8&
225: (t3) g8<(a64,b)&b..&b8&(b,a)&(a,g)a4&(a8,>a)(a32,<a)&a.
226: (t3) bb+bb+bage-de-dcdc>ba gaf+ge4.<g8g4&
227: (t3) (g4,>g)&(g4,<g)g8g8g8>(e32,f+)&f+.&  ¯6<d8d4d4_6r8
228: (t3) (c32,f+)&f+.a8& {af+<de}2(e32,f+)&f+.&e&de4&
229: (t3) (e4,>e)<l24|:4bg&af+d&e:|l16b+(a,g)&g&a(a32,b)&b.&b
4(a32,b)&b.&
230: (t3) (b4,a+)&a+8b8&(b4,>b)<r4 f+gf+gf+ed+dc+c-c+>bar8.
231: /   F  part(トラック2と3全く同じ)
232: (t2) @7518@k194q8v14p3o5@s6@m20@h44
233: (t2) r(g32,f+)&f+16.(g+32,f+)&f+..e4c+(f+32,g+)&g+16.&
234: (t2) g+(b32,<c+)&c+16.&c+2&c+>{bg+}
235: (t2) {gf+ec+f+c+ef+}2{g+e}{f+g+f+}{ec+>bg+}4
236: (t2) <<{gf+}ec+>bg+(g32,f+)&f+..e f+.e16g+4b<c+(c+32,e)&
e..
237: (t2) @q1c+2r{c-c+}{eg+f+e}4 q8{c+f+ec+c-ec+>bg+f+eg+f+ec
+f+}1
238: (t2) {ec+c-ec+>bg+f+eg+f+ec+f+ec+}1
239: (t2) c-2c+4.e f+g+b<c+(e32,f+)&f+..r4
240: (t2) (f+32,g+)&g+16.b<c+{f+g+}f+ec+e& ef+g+4f+4r4
241: (t2) l16rr|:3f+g+f+ec+8:||:4f+g+:|{f+g+f+}8ec+
242: (t2) l8c+>bg+(g32,f+)&f+..ec+>b g+(g32,f+)&f+..ec+2<
243:
244: (t3) @7518@k188q8v12p1o5@s6@m20@h44
245: (t3) r(g32,f+)&f+16.(g+32,f+)&f+..e4c+(f+32,g+)&g+16.&
246: (t3) g+(b32,<c+)&c+16.&c+2&c+>{bg+}
247: (t3) {gf+ec+f+c+ef+}2{g+e}{f+g+f+}{ec+>bg+}4
248: (t3) <<{gf+}ec+>bg+(g32,f+)&f+..e f+.e16g+4b<c+(c+32,e)&
e..
249: (t3) @q1c+2r{c-c+}{eg+f+e}4 q8{c+f+ec+c-ec+>bg+f+eg+f+ec
+f+}1
250: (t3) {ec+c-ec+>bg+f+eg+f+ec+f+ec+}1
251: (t3) c-2c+4.e f+g+b<c+(e32,f+)&f+..r4
252: (t3) (f+32,g+)&g+16.b<c+{f+g+}f+ec+e& ef+g+4f+4r4
253: (t3) l16rr|:3f+g+f+ec+8:||:4f+g+:|{f+g+f+}8ec+
254: (t3) l8c+>bg+(g32,f+)&f+..ec+>b g+(g32,f+)&f+..ec+2<
255: /   B と同じです
256: (t2) (f+32,g+)&g+*138r4 (f+32,g+)&g+*66d+4.>b&<{c+&d+}
257: (t2) e2.r4 (d32,e)&e..&ec+4.>a4< (c+32,d+)&d+*138r4
258: (t2) (c+32,d+)&d+*66>b4.f+4 b4.<c+&c+2 c+4.d+4ef+g+&
259: (t3) (f+32,g+)&g+*138r4 (f+32,g+)&g+*66d+4.>b&<{c+&d+}
260: (t3) e2.r4 (d32,e)&e..&ec+4.>a4< (c+32,d+)&d+*138r4
261: (t3) (c+32,d+)&d+*66>b4.f+4 b4.<c+&c+2 c+4.d+4ef+g+&
262: /   C  part(繰り返し記号が追加されています)
263: (t2) |:@q1|:g+f+4e4d+4(e32,f+)&f+16.& f+e4d+4c+>b<c+&
264: (t2) c+2&c+c+|g+(e32,f+)&f+16. ^2ref+(f+32,g+)&g+16.&:|
265: (t2) f+(c+32,d+)&d+16.& d+2red+c+& c+4d+er(f+32,g+)&g+16
.rf+&
266: (t2) f+4.(f+32,g+)&g+16.&g+4r>b& b4<c+d+rf+r(d+32,e)&e16
.&
267: (t2) e2.r>a& a4b<c+r(d+32,e)&e16.rd+& d+q8c+4>b4ag+f+
268: (t2) g+f+g+abg+ab <c+>ab<c+d+ef+(f+32,g+)&g+16.&:|
269:
270: (t3) |:@q1|:g+f+4e4d+4(e32,f+)&f+16.& f+e4d+4c+>b<c+&
271: (t3) c+2&c+c+|g+(e32,f+)&f+16. ^2ref+(f+32,g+)&g+16.&:|
272: (t3) f+(c+32,d+)&d+16.& d+2red+c+& c+4d+er(f+32,g+)&g+16
.rf+&
273: (t3) f+4.(f+32,g+)&g+16.&g+4r>b& b4<c+d+rf+r(d+32,e)&e16
.&
274: (t3) e2.r>a& a4b<c+r(d+32,e)&e16.rd+& d+q8c+4>b4ag+f+
275: (t3) g+f+g+abg+ab <c+>ab<c+d+ef+(f+32,g+)&g+16.&:|
276: /   G  part(2と3全く同じです)
277: (t2) o5g+2..(b32,<c+)&c+16.& c+2.r(e32,f+)&f+16.&
278: (t2) f+2..e d+4>bf+g+b<c+4 c+2r2 r4ef+g+4f+e& e4f+d+4e4c
+&
279: (t2) c+d+4c-4c+>g+f+ e4f+d+4{ed+}>b4 <c+4>g+f+ef+c+e&
280: (t2) ef+g+b<c+ere& ef+r2{f+g+}b <c+ere4f+g+4
281: (t2) f+4ec+>b(a32,g+)&g+..f+ ef+g+(e32,f+)&f+..{g+f+}ec+
282: (t2) e4f+c+4.r4 r{c-c+e}{c-c+ec-ef+}4g+<e{c+f+g+e}4
283: (t2) l16ec-{f+d+c-}8ef+g+f+{c+c-c+}8ef+ec-c+f+
284: (t2) {g+f+ec-c+ef+g+f+ec-c+}2g+8f+ec-c+{ef+e}8
285:
286: (t3) o5g+2..(b32,<c+)&c+16.& c+2.r(e32,f+)&f+16.&
287: (t3) f+2..e d+4>bf+g+b<c+4 c+2r2 r4ef+g+4f+e& e4f+d+4e4c
+&
288: (t3) c+d+4c-4c+>g+f+ e4f+d+4{ed+}>b4 <c+4>g+f+ef+c+e&
289: (t3) ef+g+b<c+ere& ef+r2{f+g+}b <c+ere4f+g+4
290: (t3) f+4ec+>b(a32,g+)&g+..f+ ef+g+(e32,f+)&f+..{g+f+}ec+
291: (t3) e4f+c+4.r4 r{c-c+e}{c-c+ec-ef+}4g+<e{c+f+g+e}4
292: (t3) l16ec-{f+d+c-}8ef+g+f+{c+c-c+}8ef+ec-c+f+
293: (t3) {g+f+ec-c+ef+g+f+ec-c+}2g+8f+ec-c+{ef+e}8
294:
295: /   2と3全く同じです
296: (t2) ¥8c-c+g+f+ec-{c+ef+}8ec-c+f+g+8f+e
297: (t2) c+8ed+e8d+c+c-c+d+c-c+d+c+d+
298: (t2) c+ed+c+c-c+c-c+c-c+>bg+ed+c+8
299: (t2) {f+d+ef+g+b}4<c+8{d+ef+g+}8d+ed+c+f+fed+
300: (t2) c+>b{f+ec+}8f+4r4{c+d+ef+g+b}4
301:
```

▶俺の初代X68000もメインメモリ2Mバイトになった。だが，ネジが1本余った。ま，いいか。一応動くし……。
落合　明浩(20)福島県

```
(t3) ¥8c-c+g+f+ec-{c+ef+}8ec-c+f+g+8f+e
(t3) c+8ed+e8d+c+c-c+d+c-c+d+c+d+
(t3) c+ed+c+c-c+c-c+c-c+>bg+ed+c+8
(t3) {f+d+ef+g+b}4<c+8{d+ef+g+}8d+ed+c+f+fed+
(t3) c+>b{f+ec+}8f+4r4{c+d+ef+g+b}4
/   Guitar
/    Intro
(t4)  @71o5l16q8v12@k0
(t4) v9eab<e>baea b<e>v10bav11eav12b<v13e>
(t4) v9baeab<e>ba eav10b<e>v11bav12ev13a
(t4) v9b<e>baeab<e> bav10eav11b<e>v12bv13a
(t4) v9eab<e>baea b<e>v10bav11eav12b<v13e>
(t4) |:|:v9eab<e>baea b<e>v10bav11eav12b<v13e>
(t4) v9baeab<e>ba eav10b<e>v11bav12ev13a
(t4) v9b<e>baeab<e>bav10eav11b<e>v12bv13a
(t4) v9eab<e>bv10acv11dev12g<cv13deg>v9ea
(t4) v9b<e>baeab<e>bav10eav11b<e>v12bv13a
(t4) v9eab<e>baeab<e>v10bav11eav12b<v13e>
(t4) v9baeab<e>ba eav10b<e>v11bav12ev13a
(t4) v9b<e>baf+a<df+>df+v10b<v11dv12gdv13c-v14d>:||
(t4) r1v9|:14c:|
/   A part
(t4) @76v11l8o3p3 b&
(t4) |:|:bbbrr4.<d& dddrr4.d& dddrr4.e&
(t4) eec+>a|<d>bgb&:||o4gar>b&:|o4g4a4
/   B part
(t4) |:16e:||:16d:||:16c:| f+4.g&g2 g4.a&a4.b&
/   C part
(t4) |:bbb4bbra& aaa4aare& eee4eerf+& f+f+f+4f+f+r|b&:|
(t4) e& eee4eerc& cc<c4(c8,d)&d4>d& ddd4ddrg& gggggggg&
(t4) ggggggf+& |:8f+:| f+4.g4.a4 g4.a4.a+b
/   D part
(t4) @71o5l16q8v12@k0:| r1r1 @76v11l8o4r2..g&
/   E part
(t4) g4g4>g<g>gg< g4>g<gg4>g<f+& f+4f+4>f+<f+>f+f+<
(t4) f+4>f+<f+f+4>f+<e& e4e4>e<e>ee< e4>ee<ee4d&
(t4) dddddd>d<c& cccccc>cb& bbbb>b<b>b<b b4>b<bbb>b<<a&
(t4) aaaaaa>a<a aaaaaa>a<g& gggg>g<g>g<g gggggg>g<f+& |:
15f+:|>b&
/   F part (Aと同じ)
(t4)  |:|:bbbrr4.<d& dddrr4.d& dddrr4.e& eec+>a|<d>bgb&:
||o4ga4>b&:|o4g4a4
/   B part
(t4) |:16e:||:16d:||:16c:| f+4.g&g2 g4.a&a4.b&
/   C part(前のCと、最初と最後が違うだけ)
(t4) |:|:bbb4bbra& aaa4aare& eee4eerf+& f+f+f+4f+f+r|b&:
|
(t4) e& eee4eerc& cc<c4(c8,d)&d4>d& ddd4ddrg& gggggggg&
(t4) ggggggf+& |:8f+:| f+4.g4.a4 g4.a4.a+|b&:|b
/   G part(Aの手前6行とほぼ同じ)
(t4) @71o5l16q8v12
(t4) |:3v9eab<e>baeab<e>v10bav11eav12b<v13e>
(t4) v9baeab<e>ba eav10b<e>v11bav12ev13a
(t4) v9b<e>baeab<e>bav10eav11b<e>v12bv13a
(t4) |v9eab<e>bv10acv11dev12g<cv13deg>v9ea
(t4) v9b<e>baeb<e>bav10eav11b<e>v12bv13a
(t4) v9eab<e>baeab<e>v10bav11eav12b<v13e>
(t4) v9baeab<e>baeav10b<e>v11bav12ev13a
(t4) v9b<e>baf+a<df+>df+v10b<v11dv12gdv13c-v14d>:|
(t4) ¥8
(t4) v9eab<e>bv10acv11dev12g<cv13deg>v9ea
(t4) v9b<e>baeab<e>bav10eav11b<e>v12bv13a
(t4) v9eab<e>baeab<e>v10bav11eav12b<v13e>
(t4) v9baeab<e>baeav10b<e>v11bav12ev13a
(t4) v9b<e>baf+a<df+>df+v10b<v11dv12gdv13c-v14d>
/*        トラック4をコピーしてエディットすれば、楽だよ
/    Intro
(t5)  @71o5l16q8v10@k5 rr
(t5) v7eab<e>baea b<e>v8bav9eav10b<v11e>
(t5) v7baeab<e>ba eav8b<e>v9bav10ev11a
(t5) v7b<e>baeab<e> bav8eav9b<e>v10bv11a
(t5) v7eab<e>baeab<e>v8ba @72l8v12o3@s3@m6@h24p1 e
(t5) |:|:|:b<egb<cd>bg af+d|gecf+d>:|o4c&c4.>e
(t5) b<egb<p3cd>bg af+dgecf+d ec>a<d>bgf+b&
(t5) bb<d4d4.|>e:|o4e& |eegg&g2& g2.r
/   A part
(t5) @76v11l8o3p2@m e&
(t5) |:|:eeerr4.f+& f+f+f+rr4.g& gggrr4.a&
(t5) ar_3<ec+|>a<d>b~3e&:|~3o4|c+dr>e&:|o4c+4d4
/   B part
(t5) |:6r1:|>b4.<c&c2 c4.d&d4.e&
/   C part
(t5) |:eee4eerd& ddd4ddr>a& aaa4aarb& bbb4bbr<|e&:|
(t5) o3a& aaa4aarr r2<(c8,d)&d4>g& ggg4ggr<c& cccccccc&
(t5) ccccccc>b& |:8b:| b4.<c4.d4 c4.d4.d+e
/   D part
(t5) @72l8v12o3@m6@h24p3:|o4eegg&ggab& bb<dd{r}>af+dge}2
(t5) {c>ab}4&b4&(b4.,>>b)@76@mo4v11p1c&
/   E part
(t5) c4c4>c<c>cc< c4>c<cc4>cb& b4b4>b<b>bb<
(t5) b4>b<bb4>b<a& a4a4>a<a>aa< a4>aa<aa4g&
(t5) gggggg>g<f& ffffff>f<g& gggg>g<g>g<g g4>g<ggg>g<<d&
(t5) dddddd>d<d dddddd>d<c& cccc>c<c>c<c cccccc>cb& |:15
b:|p1e&
/   F part (Aと同じ)
(t5) |:|:eeerr4.f+& f+f+f+rr4.g& gggrr4.a&
(t5) ar_3<ec+|>a<d>b~3e&:|~3o4|c+dr>e&:|o4c+4d4
/   B part
(t5) |:6r1:|>b4.<c&c2 c4.d&d4.e&
/   C part(前のCと、最初と最後が違うだけ)
(t5) |:|:eee4eerd& ddd4ddr>a& aaa4aarb& bbb4bbr<|e&:|
(t5) o3a& aaa4aarr r2<(c8,d)&d4>g& ggg4ggr<c& cccccccc&
(t5) ccccccc>b& |:8b:| b4.<c4.d4 c4.d4.d+|e&:|e
/   G part(Aの手前3行とほぼ同じ)
(t5) @72l8v12o3@m6@h24p3
(t5) |:|:b<egb<cd>bg af+d|gecf+d>:|o4c&c4.>e
(t5) b<egb<p3cd>bg af+dgecf+d ec>a<d>bgf+b& bb<d4d4.>e:|
(t5) |:b<egb<cd>bg |af+dgecf+d>:|¥8o4af+dc&c4.>e
(t5) b<egb<p3cd>bg af+dgecf+d ec>a<d>bgf+b& bb<d4d4.>e
/   Chord
/    Intro
(t6) @73q8v11@k0l8 r1r1r1r2..o1e
(t6) |:|:|:b<egb<cd>bg af+d|gecf+d>:|o2c&c4.>e
(t6) b<egb<cd>bg af+dgec|f+d ec>a<d>bgf+e& eef+4g4.e:|o3
f+d
(t6) ec>a<d>bgf+e& eef+4g4.a& |aab<c&c2 c2.r4
/   A part
(t6) @74v13q8l8  |:15r1:| r2o2e4f+4
/   B part
(t6) f+2&f+a4. f+2f+2 e2&eb+4. g2g2 a2&a<d4.>
(t6) a2a2 d4.e&e2 e4.f+&f+4.g&
/   C part
(t6) |:ggg4ggrf+& f+f+f+4f+f+re& eee4eera&
(t6) aaa4aar|g&:|e& eee4eere& eee4eerf+& f+f+f+4f+f+rg&
(t6) ggg4ggrg& ggg4ggrf+& f+f+f+4f+f+rf+
(t6) d4.e4.f+4 e4.f+4.gg
/   D part
(t6) @73v11l8o1:| o2aab<c&ccde& eef+g{raf+dge}2
(t6) {c>ab}4&b2@74v13o2re&
/   E part
(t6) e4e4rer4 e4ree4rf+& f+4f+4rf+r4 f+4r4f+4re&
(t6) e4e4rbr4 e4r4b4rd& d<ddddd4>d& d<ccccc4>>b&
(t6) b4b4rbrb rbrbrbra& a4a4rara <<ddddddd>e&
(t6) eee4rere e4rererd+& |:8d+:|
(t6) d+l48~3<<<gfedc>b agfedc>bagfed @v0l8r4r16@77o4v12b
8.&
/   F part
(t6) @s8@m40@h1b4.&@m~3(b8,>b)r2
(t6) r1r1@75l8@k194q8v12p3o4l16gf+ec+c+c+ef+gef+4.
(t6) |:11r1:| @k0@74v13q8l8r2o2e4f+4
/   B part
(t6) f+2&f+a4. f+2f+2 e2&eb+4. g2g2 a2&a<d4.>
(t6) a2a2 d4.e&e2 e4.f+&f+4.g&
/   C part(前のCと、最初と最後が違うだけ)
(t6) |:|:ggg4ggrf+& f+f+f+4f+f+re& eee4eera&
(t6) aaa4aar|g&:|e& eee4eere& eee4eerf+& f+f+f+4f+f+rg&
(t6) ggg4ggrg& ggg4ggrf+& f+f+f+4f+f+rf+
(t6) d4.e4.|f+4 e4.f+4.gg&:|d4 c4.d4.r4
/   G part
(t6) @73v10l8o1
(t6) |:|:b<egb<cd>bg af+d|gecf+d>:|o2c&c4.>e
(t6) b<egb<cd>bg af+dgecf+d ec>a<d>bgf+e& eef+4g4.e:|
(t6) |:|:b<egb<cd>bg |af+dgecf+d>:|¥8o2af+dc&c4.>e
(t6) b<egb<cd>bg af+dgecf+d ec>a<d>bgf+e& eef+4g4.e
/    Intro
(t7)  @74v12q8o3  @k3l8
(t7) |:7r1:| |:o2r4.c2>b b4r2. r1r2.rb& bb<d4d4.>b b1<
(t7) |:d4.c4.d4 |>b1< d4.c2>b b1<:|
(t7) o2ec>a<d>bgf+b& bb<d4d4.e& |eegg&g2 r1
/   A part
(t7) l8v13p1|:15r1:| r2o2v13c4d4
/   B part
(t7) d2&df+4. d2d2 c2&cg4. e2e2 f+2&f+a4.
(t7) f+2f+2 >b4.<c&c2 c4.d&d4.e&
/   C part
(t7) |:eee4eerd& ddd4ddrc& ccc4ccrf+&
(t7) f+f+f+4f+f+r|e&:|c& ccc4ccrc& ccc4ccrd& ddd4ddre&
(t7) eee4eere& eee4eerd& ddd4ddrd
(t7) >b4.<c4.d4 c4.d4.d+e
/   D part
(t7) r1r1r1v12p3:|
(t7) eegg&ggab& bb<dd{r}af+dge}2 {c>ab}4&b2v13p1r<c&
/   E part
(t7) c4c4rcr4 c4rcc4rd& d4d4rdr4 d4r4d4rc&
(t7) c4c4rgr4 c4r4g4r>b& b<bbbbb4>a& a<aaaaa4>g&
(t7) g4g4rgrg rgrgrgrf+& f+4f+4rf+rf+ <aaaaaaac&
(t7) ccc4rcrc c4rcrcr>b& |:8b:|
(t7) bl48~3<<<edc>bag fedc>bagfedc>b @v0l8r4r8.@77o4v10p
3b8.&
/   F part
(t7) @s8@m40@h1b4.&@m~3(b8,>b)r2
(t7) r1r1@75l8@k188q8v10p3o4l16gf+ec+c+c+ef+gef+4
(t7) |:11r1:| @k3@74v13q8l8r2o2c4d4
/   B part
(t7) d2&df+4. d2d2 c2&cg4.  e2e2 f+2&f+a4.
(t7) f+2f+2 >b4.<c&c2 c4.d&d4.e&
/   C part(前のCと、最初と最後が違うだけ)
(t7) |:|:eee4eerd& ddd4ddrc& ccc4ccrf+&
(t7) f+f+f+4f+f+r|e&:|c& ccc4ccrc& ccc4ccrd& ddd4ddre&
(t7) eee4eere& eee4eerd& ddd4ddrd
(t7) >b4.<c4.|d4 c4.d4.d+e&:|>a4 g4.a4.r4
/   G part
(t7) v11p3 |:|:o1 b1 <d4.c4.d4> |b1 <d4.c2>b:|
(t7) o2ec>a<d>bgf+b& bb<d4d4.>b:|
(t7) v11p3    |:o1 b1 <d4.c4.d4> |b1 ¥8<d4.c2>b:|
(t7) o2ec>a<d>bgf+b& bb<d4d4.>b
/    Intro
(t8) r*2 @74v12q8o3  @k-3l8
(t8) |:7r1:| |:o1r4.g2g g4r2. r1 r2.rg& gga4b4.g g1
(t8) |:a4.g4.a4 |g1 a4.g2g g1:|
(t8) o2r_2ec>a<d>bg~2g& gga4b4.<c& |ccdd&d2 r1
/   A part
(t8) v13p2|:15r1:| r2o1g4a4
```

▶はがゆい。あと一歩のところで蚊を逃してしまった。ところでアンケートハガキの住所欄が少し広くなりましたね。最初，気がつかなくて思わずのびのびとでかい字で書いてしまった。はは。

林 大介(17)神奈川県

```
/   B part
(t8) a2&a<d4.> a2a2 g2&g<e4. c2c2 d2&df+4.
(t8) d2d2 >a4.g&g2 g4.a&a4.b&
/   C part
(t8) |:bbb4bbra& aaa4aara& aaa4aar<d&
(t8) ddd4ddr>|b&:|a& aaa4aara& aaa4aarb& bbb4bbr<c&
(t8) ccc4ccrc& ccc4ccr>b& bbb4bbrb
(t8) a4.g4.a4 g4.a4.a+b
/   D part
(t8) r1r1r1v12p3:|
(t8) o2ccde&eef+g& ggab{rr_2af+dg}2 {ec>a}4b2~2v13p2rb&
/   E part
(t8) b4b4rbr4 b4rbb4ra& a4a4rar4 a4r4a4ra&
(t8) a4a4r<er4> a4r4<e4>rg& g<ggggg4>f+& f+<f+f+f+f+f+4>e&
(t8) e4e4rere rererd& d4d4rdrd <|:7f+:|>g&
(t8) ggg4rgrg g4rgrgrf+& |:8f+:|
(t8) f+l48<<gfedc>b agfedc>bagfed @v0l8r2
/   F part
(t8) |:15r1:| v13q8l8r2o1g4a4
/   B part
(t8) a2&a<d4.> a2a2 g2&g<e4. c2c2 d2&df+4.
(t8) d2d2 >a4.g&g2 g4.a&a4.b&
/   C part(前のCと、最初と最後が違うだけ)
(t8) |:|:bbb4bbra& aaa4aara& aaa4aar<d&
(t8) ddd4ddr>|b&:|a& aaa4aara& aaa4aarb& bbb4bbr<c&
(t8) ccc4ccrc& ccc4ccr>b& bbb4bbrb
(t8) a4.g4.|a4 g4.a4.a+b&:|f+4 e4.f+4.r4
/   G part
(t8) v11p3 |:o1 |:g1 a4.g4.a4 |g1 a4.g2g :|
(t8) o2r_2ec>a<d>bg~2g& gga4b4.g:|
(t8) v11p3    o1 |:g1 a4.g4.a4 |g1 ¥8a4.g2g :|
(t8) o2r_2ec>a<d>bg~2g& gga4b4.g


/ Hi-hat
/    Intro
(t9) o0v9l4@F4q8 |:4rrer:| l8|:6|:8e:|:| |:6e:|rr
(t9) rer2. |:3|:8e:|:| rerrrerr |:|:8e:|:| r1r1r1r2..f
/   A part
(t9) |:31e:|r r|:23e:| |:4e:|r2 r|:7e:|
(t9) |:23e:|r r|:23e:| |:4e:|r2
/   B part
(t9) rr|:6e:| |:56e:|
/   C part
(t9) |:63f:|r l4|:24g:|r1r1l8
/   D part
(t9) |:3r2e4r4:|r1 r|:7e:| r4.eeer4 r1
(t9) r2.g4 |:48e:| r1r1 |:13e:|rrr r1
/   E part
(t9) l4|:60g:|rrrgl8
/   F part
(t9) |:31e:|r r|:23e:| |:4e:|r2 r|:30e:|r r|:29e:|r4
/   B part
(t9) rr|:6e:| |:56e:|
/   C part
(t9) |:|:63f:|r l4|:24g:|r1|r1l8:|l8r2rfrr
/   G part
(t9) g4r4ggr4 |:50g:|r2g4 r4|:54g:| rgr2g4
(t9) r4|:22g:| ¥8|:6g:|rr |:24g:| rgr2g4



/ Bass Drum
/    Intro
(t10) o0 v9l8@F4q8 |:c4r2. c4r2rc :|
(t10) |:c4r2. c4r2rc c4|r2. c4rc4rrc:|r2rc rcr2.
(t10) |:3c4r4ccr4:| c4rc4crc |:c4r4ccr4:| rcc4ccrc rcc4r4.c
(t10) r4.c4.r4 r2..c
/   A part
(t10) |:|:3rcr2rc:| |rcr4.crc:|rcr4c4rc
(t10) |:|:3rcr2rc:| |rcr4.ccc:|rcrcc4cc
/   B part
(t10) |:6c4r4ccr4:|  |:c4rc4.|r4:|c4
/   C part
(t10) |:32c4:| rc4.cc4c& |:3cr4.ccr|c& ccr4ccrc&:|r |:c4rc4.r4:|
/   D part
(t10) c4r2. c4r2rc c4r2. c4rc4rrr c4r2. r1 r2..c rcr2rc
(t10) |:3c4r4ccr4:| c4rc4crc |:c4r4ccr4:| rcc4ccrc rcc4r4c4
(t10) |:7c4:|r4 {rrc}4&c4r4rc
/   E part
(t10) |:3r4r4ccr4 c4r4ccrc:| c4r4ccrc |:3rcr4ccrc:|
(t10) |:rcr4ccr4 c4r4ccrc:| rcr4ccrc rcc4rccc
/   F part
(t10)  |:|:3rcr2rc:| |rcr4.crc:|rcr4c4rc
(t10) |:|:3rcr2rc:| |rcr2cc:|rcrcrcr4
/   B part
(t10) |:6c4r4ccr4:|  |:c4rc4.|r4:|c4
/   C part
(t10) |:32c4:| rc4.cc4c& |:3cr4.ccr|c& ccr4ccrc&:|r |:c4rc4.r4:|
(t10) |:32c4:| rc4.cc4c& |:3cr4.ccr|c& ccr4ccrc&:|r |:c4rc4.|r4:|cc
/   G part
(t10) |: c4r4ccr4   c4.c4cr4 |c4r4ccr4 c4rc4.rc:|ccrc4crc rcr2rc
(t10) |:3c4r4ccr4:| c4.c4cr4  c4r4ccr4 c4rc4cr4 c4r4ccrc rcr2rc
(t10) |:3c4r4ccr4:| ¥8c4.c4{cc}r4 |:c4r4ccr4:| c4r4ccrc rcr4c4.c


/ Snare Drum
/    Intro
(t11) o0 l4@F4v9q8 |:7rd:|r8d16dr16 |:7rd:|r2 |:7rd:|dr
(t11) |:12rd:| dr8dr8d r2dr r1 r2r8d8d16d16r8
/   A part
(t11)  |:7rd:|dd |:7rd:|r8d4. |:7rd:|dr |:7rd:|r2
/   B part
(t11) |:15rd:|r4.d8&
/   C part
(t11) |:|:7rd:|rd8d8 :| |:13rd:|r8d8d rd4r2
/   D part
(t11) ddrd |:5rd:|rr |:4rd:|r1r1 |:12rd:|
(t11) |:v8d64v9d8...r8:|v8d64v9d8...
(t11) r2dr8d8& d8rdrd8& d8rd8&d12d4d6 r2.d
/   E part
(t11) |:30rd:| drdr
/   F part
(t11)  |:|:7rd:|dr |:7rd:| |r8d4.:|dd
/   B part
(t11) |:15rd:|r4.d8&
/   C part
(t11) |:|:|:7rd:|rd8d8 :| |:14rd:| |rdr4.d8&:|r1
/   G part
(t11) |:|:15rd:|dr:|
(t11) |:6rd:| ¥8|:9rd:|rr
/ TomTom
/    Intro
(t12) o1v9l8@F4q8 |:3r1:|r2.e4 |:7r1:|
(t12) r4.f4edc |:8r1:| rcc4r2 l16fedcf8fedcr4.l8
/   A & B part
(t12)  |:24r1:|
/   C part
(t12) |:15r1:|r2.fe
/   D part
(t12) r1 l16ffe8d4r4e8d8 r1 r2.e8d8 r1 ffe8d4r4r8cc
(t12) c8d8|:3dd|d8:|r8 r4|:3f64f..:|e8d8c8 |:8r1:| r1
(t12) r2{rrfedd}2 {dcr}4r2.
/   E & F part
(t12) |:32r1:|
/   B part
(t12) |:8r1:|
/   C part
(t12) |:|:15r1:| |r2r8c8c4:|r1
/   G part
(t12) |:3|:8r1:|:|

/ Crush Cymbal
/    Intro
(t13) q8o0v9l8|:4r1:| a2r2 r1r1 a4.a2a8 a2r2 r1
(t13) r2..a8& a4a4a2 a2r2 r1r1 a4.a2a a2r2 r1 r2..a&
(t13) a4a4a4aa& a4.a4.r4 r1
/   A part
(t13) |:3r1:|r2..a& a2r2 r1r1 r2a8a4a& a2r2 r1r1r2..a&
(t13) a2r2 r1r1 r2a4a4
/   B part
(t13) a2r2 |:5r1:| |:a4.a2|r:|a&
/   C part
(t13) |:7r1:|r2..a& a4r2. |:5r1:| a4.a4a8a4 a4.a4.r4
/   D part
(t13) |:3r1:| r4a2r4 a2r2 r1 r2..a& a2r2 a2r2 r1r1
(t13) r4.a2a a2r2 r1 r2..a& a4a4a4.a& a4.a2a&
(t13) a4.a2r {rra}4&a4r4.a&
/   E part
(t13) a2r2|:5r1:|r2..a& a2r4.a& a2r2 |:r2..a& a2r2:|r2..a& a2r2a2a4.a&
/   F part ( '=, A)
(t13) |:3r1:|r2..a& a2r2 r1r1 r2a8a4a& a2r2 r1r1r2..a&
(t13) a2r2 r1r1 r2.a4
/   B part
(t13) a2r2 |:5r1:| |:a4.a2|r:|a&
/   C part
(t13) |:|:7r1:|r2..a& a4r2. |:5r1:| a4.a4.a4 a4.|a2a8&:|a4.a4
/   G part
(t13) r4a2a4& a4r2. |:5r1:| r4a4a4r4
(t13) a2r2 |:5r1:| r2..a& a4a4a4r4
(t13) a2r2 |:r1:| ¥8r2.a4 r1r1 r2..a& a4a4a4r4


/ Chord strings
/    Intro
(t14) q8 |:22r1:|
/   A part
(t14) |:15r1:| r2o5@F4v9l8eef+f+
/   B part
(t14) |:16f+:| |:16e:| |:16a:| dddeeeee eeef+f+f+f+r
/   C part
(t14) |:7r1:| r2..e |:15e:|d |:7d:|e |:15e:|d
(t14) |:8d:| d4.e4.f+4 e4.f+4.gg
/   D part
(t14) |:16r1:| r1r1r2..c
/   E part
(t14) |:15c:|@F2b |:15b:|@F4e |:15e:|d |:7d:|c
(t14) |:7c:|@F2b |:8b:| @F4v11|:7e:|d |:8d:|
(t14) |:7a:|g |:15g:|f+ |:6f+:|rr r1
/   F part (=A)
(t14) |:15r1:| r2o5@F4v9l8eef+f+
/   B part
(t14) |:16f+:| |:16e:| |:16a:| dddeeeee eeef+f+f+f+r
/   C part(前のCと、最初と最後が違うだけ)
(t14) |:|:7r1:| r2..e |:15e:|d |:7d:|e |:15e:|d
(t14) |:8d:| d4.e4.f+4 e4.f+4.r4:|
/   G part
(t14) |:3|:8r1:|:|


/    Intro
```

▶10月号90ページの日高のり子さんというのが，たいへん気になるのですが。
山口　英斉(16)広島県

```
(t15) r*2q8|:22r1:|
/   A  part
(t15) |:15r1:| r2o5@F4v918ccdd
/   B  part
(t15) |:16d:| |:16c:| |:16f+:| @F2bbb@F4ccccc cccddddr
/   C  part
(t15) |:7r1:| r2..c |:15c:|@F2b |:7b:|@F4c |:15c:|@F2b
(t15) |:8b:| b4.@F4c4.d4 c4.d4.d+e
/   D  &  E  part
(t15) |:35r1:|
/   F  part (=A)
(t15) |:15r1:| r2o5@F4v918ccdd
/   B  part
(t15) |:16d:| |:16c:| |:16f+:| @F2bbb@F4ccccc cccddddr
/   C  part(前のCと、最初と最後が違うだけ)
(t15) |:|:7r1:| r2..c |:15c:|@F2b |:7b:|@F4c |:15c:|@F2b
(t15) |:8b:| b4.@F4c4.d4 c4.d4.r4:|
/   G  part
(t15) |:3|:8r1:|:|


/    Intro
(t16) r*1q8|:22r1:|
/   A  part
(t16) |:15r1:| r2o5@F2v918ggaa
/   B  part
(t16) |:16a:| |:16g:| @F4|:16d:| @F2f+f+f+ggggg gggaav12
@F4ef+g&
/   C  part
(t16) |:gf+4e4d4f+& f+e4d4c@F2b@F4c& c2&cc|gf+
(t16) f+2ref+g&:|f+d& d2r4.p3v9@F2g
(t16) |:24g:| |:15g:|a |:8a:| a4.g4.a4 g4.a4.a+b
/   D  &  E  part
(t16) |:35r1:|
/   F  part (=A)
(t16) |:15r1:| r2o5@F2v918ggaa
/   B  part
(t16) |:16a:| |:16g:| @F4|:16d:| @F2f+f+f+ggggg gggaav12
@F4ef+g&
/   C  part(前のCと、最初と最後が違うだけ)
(t16) |:|:gf+4e4d4f+& f+e4d4c@F2b@F4c& c2&cc|gf+
(t16) f+2ref+g&:|f+d& d2r4.p3v9@F2g
(t16) |:24g:| |:15g:|a |:8a:| a4.g4.a4 |@F4v12c@F2ab@F4c
def+g&:|g4.a4.r4
/   G  part
(t16) |:3|:8r1:|:|


(p)
```

リスト4　ARCADIA用コンフィグ・ファイル

```
/ ARCADIA       T-SQUARE             Programed by 岡田　一彦

.o0d = DSNAR.pcm ,v62
.o0c = KICK3.pcm ,v32
.o1f = HLT4.pcm ,v56
.o1e = HLT3.pcm ,v56
.o1d = HLT2.pcm ,v56
.o1c = HLT1.pcm ,v56
.o0g = RDBLC1.pcm,v80
.o0e = HHC.pcm
.o0f = HO1.pcm ,p3,v75

.o0a = CRSH0.pcm ,v60


.o5c = STR.pcm ,p12,v12
.o5d = .o5c,p2
.o5d+= .o5c,p3
.o5e = .o5c,p4
.o5f+= .o5c,p6
.o5g = .o5c,p7
.o5a = .o5c,p9
.o5a+= .o5c,p10
.o5b = .o5c,p11
```

リスト5　ARCADIA用カウンタ表示

```
 1:000084C0 00000000   2:000084C0 00000000   3:000084DA 00000000   4:000084C0 00000000
 5:000084C0 00000000   6:000084C0 00000000   7:000084C0 00000000   8:000084C2 00000000
 9:000084C0 00300000  10:000084C0 00000000  11:000084C0 00000000  12:000084C2 00000000
13:000084C0 00000000  14:000084C0 00000000  15:000084C2 00000000  16:000084C1 00000000
```

リスト6　スーパーマリオブラザーズ

```
10 '「SUPER MARIO BROS.」 Programmed by N.Nakamura
20 '
30 DEFSTR A-H:DEFINT I-Z:CLS 0:TEMPO 0
40 DEFUSR=&HB000:DIM V(4,10)
50 DEFFNV$(N,V(0,0))=USR(CHR$(N)+MKI$(VARPTR(V(0,0))))
60 FOR K=1 TO 8:FOR J=0 TO 10:FOR I=0 TO 4:READ V(I,J):NEXT:NEXT
70 V$=FNV$(K,V):NEXT
80 '
90  ' NO.1  VIB
100 DATA  44, 15,  2,  0,  0,  0,  0,  0,  0,  3,  0
110 DATA  24, 14,  0,  7, 15, 57,  1, 12,  3,  0,  0
120 DATA  24, 10,  0,  7, 15,  0,  1,  4,  0,  0,  0
130 DATA  26, 14,  0,  6, 15, 57,  1,  4,  0,  0,  0
140 DATA  26,  8,  0,  6, 15,  5,  2,  1,  0,  0,  0
150 ' NO.2  BASS
160 DATA  32, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
170 DATA  31,  7,  7,  9,  2, 29,  3,  6,  7,  0,  0
180 DATA  31,  6,  6,  9,  1, 47,  3,  5,  7,  0,  0
190 DATA  26,  9,  6,  9,  1, 29,  2,  0,  7,  0,  0
200 DATA  31,  8,  4,  9,  3,  0,  2,  1,  7,  0,  0
210 ' NO.3  TRUMPET
220 DATA  58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
230 DATA  16,  5, 10,  0,  0, 27,  1,  1,  0,  0,  0
240 DATA  13, 10,  1, 10, 15, 63,  1,  2,  0,  2,  0
250 DATA  15, 10,  0, 10,  0, 47,  0,  1,  0,  0,  0
260 DATA  20, 10,  0, 10,  0,  0,  1,  1,  0,  0,  0
270 ' NO.4  CLAP
280 DATA  57, 15,  3,  0,200,127,  0,  7,  0,  3,  0
290 DATA  26, 28,  0,  3,  6,  9,  0,  5,  0,  0,  0
300 DATA  29, 20,  0,  3, 15, 15,  0,  2,  0,  1,  0
310 DATA  31, 11,  0,  3, 15, 34,  0,  0,  0,  0,  0
320 DATA  31, 14,  0,  9, 15,  2,  0,  1,  0,  3,  0
330 ' NO.5  WOOD BLOCK
340 DATA  27, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
350 DATA  31, 18,  0, 10, 15, 19,  0,  2,  0,  2,  0
360 DATA  31, 26,  0, 10, 15, 31,  0,  2,  3,  0,  0
370 DATA  31, 22,  0, 10, 15, 47,  0,  2,  0,  3,  0
380 DATA  30, 20,  0, 10, 15,  0,  0,  2,  4,  0,  0
390 ' NO.6  BASS DRUM
400 DATA  37, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
410 DATA  31, 22,  0, 15, 15,  5,  0,  0,  0,  1,  0
420 DATA  31, 17, 18, 15, 10,  0,  0,  1,  0,  0,  0
430 DATA  31, 17, 18, 15, 10,  0,  0,  1,  0,  0,  0
440 DATA  31, 17, 18, 15, 10,  0,  0,  1,  0,  0,  0
450 ' NO.7  HI-HAT
460 DATA  60, 15,  0,  0,  0,  0,  0,  0,  0,  1,  0
470 DATA  31, 31,  5,  5,  2,  0,  0, 15,  1,  0,  0
480 DATA  31, 19, 31, 15,  5,  0,  0,  9,  7,  0,  0
490 DATA  31, 11, 31, 15, 10,  4,  0, 15,  3,  0,  0
500 DATA  31, 22, 31, 15,  8,  0,  0, 15,  3,  0,  0
510 ' NO.8  SNARE
520 DATA  60, 15,  0,  0,  0,  0,  0,  0,  0,  1,  0
530 DATA  31,  0,  0,  4,  0,  0,  0, 15,  0,  2,  0
540 DATA  31, 15, 14,  6,  6,  0,  0,  1,  0,  0,  0
550 DATA  31, 25,  0,  0, 15,  0,  0,  1,  0,  1,  0
560 DATA  31, 16, 15, 10, 15,  0,  0,  1,  0,  0,  0
570 '
580 GOTO 720
590 '
600 LABEL"P"
610 'Z$=":[T112R8T224R8]"
620 FOR I=0 TO 10:PLAY A(I);:NEXT:PLAY ":";
630 PLAY B(0);:FOR I=1 TO 10:PLAY A(I);:NEXT:PLAY ":";
640 FOR I=0 TO 10:PLAY C(I);:NEXT:PLAY ":";
650 PLAY D(0);:FOR I=1 TO 10:PLAY C(I);:NEXT:PLAY ":";
660 FOR I=0 TO 10:PLAY E(I);:NEXT:PLAY ":";
670 FOR I=0 TO 10:PLAY G(I);:NEXT:PLAY ":";
680 FOR I=0 TO 10:PLAY H(I);:NEXT:PLAY ":";
690 FOR I=0 TO 10:PLAY F(I);:NEXT:PLAY Z$
700 RETURN
710 '
720 A(0)="T150I3Q8V15O5L8P2K0
730 B(0)="I3Q8V13O5L8P1K6 R32
740 C(0)="I3Q8V15O4L8P1K0
750 D(0)="I3Q8V13O4L8P2K6 R32
760 E(0)="I2Q8V14O3L8P3K0
770 F(0)="I5Q8V13O5L8K0
780 G(0)="I6Q8V15O2L8P3K0
790 H(0)="Q8V15O3L8P3K0
800 '
810 A(1)="[ EERERCERG4R2.
820 C(1)="[ F+F+RF+RF+F+RB4R4G4R4
830 E(1)="[ DDRDRDDR>G4<R4G4R4
`40 F="P3I4V15O6C4I5V13O5":F(1)="[ P2FFRFRFFRF4R4P1C4"+F
```

```
850 G(1)="[ CCRCRCCRCRRRR>C<GC
860 H(1)="[ I8CCI7CI8CI7CI8CCI7CI8CI7CCCCCCC
870 '
880 A(2)="CR4<G4.ERRARB4A+ARG6>E6G6ARFGRERCD<B4.>":A(2)=A(2)+A(2)
890 C(2)="ER4C4.<GRR>CRD4C+CRC6G6B6>CR<ABRAREFD4.":C(2)=C(2)+C(2)
900 E(2)="GR4E4.CRRFRG4F+FRE6>C6E6FRDERCR<ABG4.":E(2)=E(2)+E(2)
910 F1="P2FP1C":F2="P2FFP1CC":F(2)="L4"+STRING$(7,F1)+"P2F"+F:F(2
)=F(2)+F(2)
920 G(2)=STRING$(16,"CRRR")
930 H(2)=STRING$(7,"I7CRI8CR")+"I7CI8CC4":H(2)=H(2)+H(2)
940 '
950 A(3)="R4GF+FD+RER<G+A>CR<A>CDR4GF+FD+RER>CRCCR4.< R4GF+FD+RER
<G+A>CR<A>CDR4D+R4DR4C4R2.
960 C(3)="R4>ED+D<BR>CR<EFGRCEFR4>ED+D<BR>CRFRFFR4. R4ED+D<BR>CR<
EFGRCEFR4G+R4FR4E4R2.
970 E(3)="C4RG4R>C4<F4R>CC4<F4C4RE4RG>CR2.<G4 C4RG4R>C4<F4R>CC4<F
4C4G+R4A+R4>C4<RGG4C4
980 F(3)="L8"+STRING$(6,F2)+"P3V15I3RGRGGR"+F+STRING$(4,F2)+"P2R4
FR4FR4FR4FFR"+F
990 G(3)=STRING$(6,"CRRR")+"RCRCCRCR"+STRING$(4,"CRRR")+"RRCRRCRR
 CRRRRRCR
1000 H="I7CCI8CI7C":H(3)=H+H+H+H+H+H+"CI8C4CC4I7CR"+H+H+H+H+"CRI8
C4I7CI8C4RC4RCC4I7C4
1010 '
1020 A(4)=A(3)
1030 C(4)=C(3)
1040 E(4)=E(3)
1050 F(4)=F(3)
1060 G(4)=STRING$(6,"CRRR")+"RCRCCRCR"+STRING$(4,"CRRR")+"RRCRRCR
R CRRRR>C<GC
1070 H(4)=H(3)
1080 '
1090 A(5)="I1CCRCRCD4ECR<AG4R4>CCRCRCDE&E2R2 CCRCRCD4ECR<AG4R4>I3
EERERCERG4R2.
1100 C(5)="I1A-A-RA-RA-B-4GEREC4R4A-A-RA-RA-B-G&G2R2 A-A-RA-RA-B-
4GEREC4R4I3F+F+RF+RF+F+RB4R4G4R4
1110 E(5)="<A-4R>E-4RA-4G4RC4R<G4A-4R>E-4RA-4G4RC4R<G4A-4R>E-4RA-
4G4RC4R<G4>DDRDRDDR>G4<R4G4R4
1120 F(5)="L4"+STRING$(7,F1)+"P2F"+F+STRING$(4,F1)+"P2L8FFRFRFFRF
4R4P3C4"+F
1130 G(5)="R1R1R1R1R1R1CCRCRCCRCRRRR>C<GC
1140 H(5)="I7"+STRING$(24,"CR")+"I8CCI7CI8CI7CI8CCI7CI8CI7CCCCCCC
1150 '
1160 A(6)=A(2)
1170 C(6)=C(2)
1180 E(6)=E(2)
1190 F(6)=F(2)
1200 G(6)=G(2)
1210 H(6)=H(2)
1220 '
1230 A(7)="ECR<G4.G+4A>FRF<A4R4B6>A6A6A6G6F6ECR<AG4R4> ECR<G4.G+4
A>FRF<A4R4B>FRFF6E6D6C2R2
1240 C(7)=">C<ARE4.E4F>CRC<F4R4G6>F6F6F6E6D6C<ARFE4R4 >C<ARE4.E4F
>CRC<F4R4G>DRDD6C6<B6GEREC2
1250 E(7)="C4RF+G4>C4<F4F4>CC<F4D4RFG4B4G4G4>CC<G4 C4RF+G4>C4<F4F
4>CC<F4G4RGG6A6B6>C4<G4C2
1260 F(7)="L8"+STRING$(7,F2)+"P2FF"+F:F(7)=F(7)+F(7)
1270 G(7)=STRING$(13,"CRRR")+"C6C6C6CRRRCRRR
1280 H(7)=STRING$(13,H)+"I8C6C6C6"+H+"CCI8CC
1290 '
1300 A(8)=A(7)
1310 C(8)=C(7)
1320 E(8)=E(7)
1330 F(8)=F(7)
1340 G(8)=STRING$(13,"CRRR")+"C6C6C6CRRRC>C<GC
1350 H(8)=STRING$(13,H)+"I8C6C6C6"+H+"CCCC
1360 '
1370 A(9)=A(5)
1380 C(9)=C(5)
1390 E(9)=E(5)
1400 F(9)=F(5)
1410 G(9)=G(5)
1420 H(9)=H(5)
1430 '
1440 A(10)=A(7)+"]
1450 C(10)=C(7)+"]
1460 E(10)=E(7)+"]
1470 F(10)=F(7)+"]
1480 G(10)=G(8)+"]
1490 H(10)=H(8)+"]
1500 '
1510 "P"
```

(善)のゲームミュージックでバビンチョ

古川もとあきソロ・ライブ行ってきました

8月6日，六本木PIT INNにて，アルバム「Sound Locomotive」の発売を記念した古川もとあき氏のソロ・ライブが行われました。同アルバムプロデューサーの是方氏のギター＆ボコーダーによる乱入あり，3分以上にわたるギター＆キーボードソロ・バトルあり，キーボーディスト光田氏によるピアニカのパフォーマンスありと，聴いてよし，見てよしの最高のひとときでした。アルバムとは違うアドリブが聴けたのはなんか得した気分。

曲目は以下のとおりでした。

1．Rain On The Hill
2．SHA-SOH
3．Cash Girl
4．Invisible Bridge
5．Symphony Of Wind
6．Sound Locomotive
7．Station Love
8．Long Long Distance
9．Grand Punk Railroad
10．Fantastic Offroader

アンコール　・Memories Of Summer Island
　　　　　　・Good Night Pony

なお，古川氏のギターは，

矩形波倶楽部　KICA-1020　¥3,000
コナミ・オールスターズ　KICA-1053～1055　¥6,300
Sound Locomotive　KICS-212　¥3,000

以上，すべてキングレコードなどで聴くことができます。

●GAME MUSIC FESTIVAL -Super live'92-
CD:PCCB-00099
ポニーキャニオン　¥2,500(税込)　10/21発売

いやー，今年初めて行ってきました。ポニーキャニオンさんが招待券くれたんでね。去る8月22，23日の2日間にわたって行われたイベントです。

初日，トップのJ.D.K.バンドは，私は最近のファルコム・ミュージックはよくわかんないし，J.D.K.が何の略か知らないし，さらにわけのわかんないアレンジが利いていたんでなんか聴きづらかった。ボーカルを聴かせたいのか，ギターを聴かせたいのか……謎。でも，元気はよかったです。聴衆を盛り上げたのは彼らですね。演奏終わってから「SEX！」と叫ぶのはJ.D.K.という略称と関係があるのかどうかは謎です。

次のGAMADELICは，この夏発売されたアルバム「RAP a de LIC」から数曲と，ウルフファング，空牙などのメジャーどころで攻めたててきました。サンプラーなどを駆使して効果音を盛り込んだサウンドとMr.Kによる生ラップはなかなか熱かったです。私はひとりで来たのも忘れて大声で歌ってました。「Walk walk like a cop! Everybody walks like a cop!」今思うと恥ずかしい，あぁ……。

初日の締めはZUNTATA。やたらにキーボードが多いのと，演奏中の顔が妙に真剣なのは毎年恒例。しかし，キーボードが多いだけあって音が厚い。演奏曲はダライアス，ダライアス2，サイバリオン，ミズバク，メタルブラック，ギャラクティックストーム，そして定番ナイトストライカー＆忍者ウォーリアーズといった，ファンなら涙ものの構成。ZUNTATAもコンピュータを駆使してライブとは思えないような複雑なサウンドを聴かせてくれました。ギャラクティックストームではブレイクでアコースティックギターのスラッシュあり，忍者ウォーリアーズではお決まりの津軽三味線の乱入あり。メロディがハッキリした曲が多いので初日のなかではいちばん安心して聴けましたね。

2日目。トップはカプコン「ALFH LYRA」。全員関西人だけあって，MCではハイレベルな駄洒落で観衆をカオスへ導いていました。演奏曲はストリートファイターIIやファイナルファイトなど。

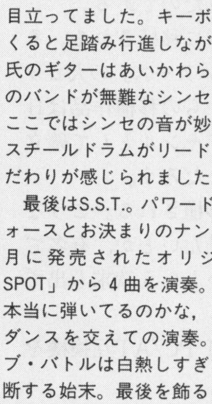
んで，次は矩形波倶楽部。ここもキーボードが目立ってました。キーボードの舟橋氏は，ノってくると足踏み行進しながら弾くのが印象的。古川氏のギターはあいかわらず歌ってましたねぇ。他のバンドが無難なシンセブラス音色を使うなか，ここではシンセの音が妙に凝っており，また途中スチールドラムがリードをとったりと，「音」にこだわりが感じられました。

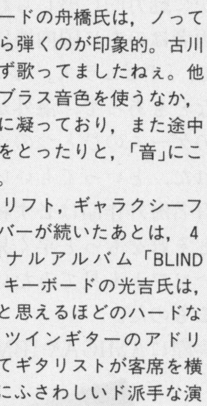
最後はS.S.T.パワードリフト，ギャラクシーフォースとお決まりのナンバーが続いたあとは，4月に発売されたオリジナルアルバム「BLIND SPOT」から4曲を演奏。キーボードの光吉氏は，本当に弾いてるのかな，と思えるほどのハードなダンスを交えての演奏。ツインギターのアドリブ・バトルは白熱しすぎてギタリストが客席を横断する始末。最後を飾るにふさわしいド派手な演奏でした。ただ，私としてはゲームミュージックのほうをもうちょっと演奏してほしかったですね。SDIはなかなか泣かせました。じーん。

お勧め度　9

終わりに

S.S.T.のオリジナルアルバム「BLIND SPOT」からR.パトレーゼ，G.ベルガーのテーマが決まったそうです。みんな応援してあげましょう。





もうひとつの情報源

海外雑誌の読み方

Akikawa Ryou
秋川　涼

最近，MacintoshやIBM PC/AT互換機を中心として，ようやく国外のパソコンが一般的になってきた。本誌でも“TREND ANALYSIS”でAMIGAのゲームを紹介しているが，わりと関心をもって読んでいただいているようだ。

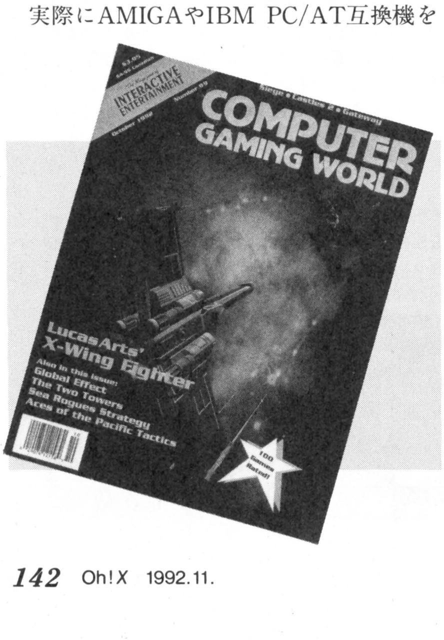
海外パソコン事情はずいぶん昔から興味をひくものとして存在している（もちろん，一部の人々にではあるが）。そもそもの主役はAPPLEIIというコンピュータ，そして，それをとりまく個性豊かなソフトウェアであった。

「ウィザードリィ」や「ウルティマ」といったゲームソフトがその最たるもの。少ない情報や日本でお目にかかれるまでのタイムラグなども微妙に影響して，実際よりもはるかに魅力的に感じられたのである。

両者はのちに国内機種に移植される。そして，「ポピュラス」「シムアース」の大ヒットなどもあって，海外からの移植作品は日本のゲームのひとつのジャンルとして確立された，といってもいいだろう。

有名海外作品は必ず移植される。こうなるとそれをもっと早く見たいとか，有名でないソフトも見てみたいという欲求も出てくるものだ。

実際にAMIGAやIBM PC/AT互換機を購入して，それらの作品を漁ってみるのもいいだろう。しかし，それには金がかかる。パソコンはいま持っている1台だけで十分だという人も多いだろう。



そこでオススメするのが，海外の雑誌で情報を収集するという方法である。

これなら，売っている店を見つければ，1,000円から3,000円程度の出費ですむ。大きなパソコンショップや輸入ソフトを扱っている店なら入手可能なので，わりと簡単に入手ルートは確保できるだろう。もし，そういう店が見つからなければ，定期購読で直輸入する手もある。

では，どのような雑誌があるのか。面白い内容のものをいくつか紹介していこう。

COMPUTER GAMING WORLD

アメリカで発売されているパソコンゲーム誌。対象機種は特に限定されていないので（とはいえ，アメリカでの勢力分布は色濃く反映されていて，やはりIBM PC用のゲームがメインになっているのだが），読むことだけを楽しむのであれば，この本がいちばんおトク。

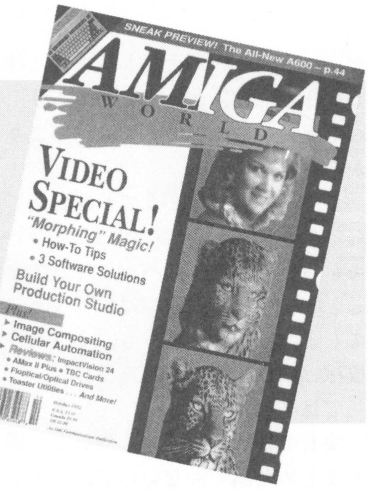
1,2ページで1本のゲームをレビューしていて，写真点数が控えめで文字量が多い。デザインもわりと地味。本誌に近い雰囲気をもっているかもしれない。囲みには対応機種，使用機種（レビューでの），値段，プロテクトの種類，デザイナーの名前，そして発売元などがまとめられている。

そのほかの記事もゲーム誌として一般的なものがほとんど揃っている。Industry News（いくつかのソフトハウスの近況を紹介），Peek，Sneak Preview（ともに新作情報），Game Hints，Replayなどなど。

読者アンケートハガキによるTop 100 Gamesも見もの。なにしろ，Top 10でなくTop 100であるから壮観である。ちなみに10月号では，1位が「Civilization」，100位がKoeiの「Genghis Khan」であった。

ゲームに関しての情報量はかなり豊富だということがおわかりいただけただろうか。値段は＄3.95で，日本で買うと1,000円から1,500円程度。

ついでにいうと，広告もゲームや入力機器を中心に多数掲載されている。前半などはほとんど1ページおきに載っていて（見開きの右側にずっと入っている）読みにくいところもあるのだが，デザイン的にレベルが高いものが多いので許そう。



AMIGA FORMAT&CU AMIGA

ともにヨーロッパのAMIGA専門誌。このほかにもAMIGA COMPUTING，AMIGA ACTION，AMIGA POWERなどがあるが，内容的には似たりよったりだ。充実度という点でこの２誌を取り上げてみた。

どちらもデザイン的にはハデハデな趣きで，一歩間違うと悪趣味になりそうだが，かろうじて留まっている。

ハデさのほかにもうひとつ，ヨーロッパの雑誌の特徴としては，ほとんどといっていいほど付録ディスクがついていることが挙げられる。3.5インチフロッピーディスクが１枚，あるいは２枚，雑誌にセロテープでじかに張りつけられているのには，あきれることさえ通り越して，むしろ豪快さを感じる。

しかし，最近ついているディスクの充実度には目をみはるものがある。グラフィック関係のアプリケーションソフトを中心に，数年前の**人気ソフトの完全版**がついているのだ。古いとはいえ，ちゃんとまだ店で売っている現役ソフトであるから，機能的には満足のいくものばかり。最近買ったソフトがついてきた日には目も当てられないというふうな状況なのである。

具体的にいうと「VISTA」「ANIMATION STATION」「MOVIE SETTER」「SCULPT 4Djr.」など。他誌の最新ゲームデモ版なんてどうでもよくなってしまうほどのラインナップである。とはいえ，ユーザーでない人々にとってはゴミ以外の何物でもないかもしれない。これはご参考までに。

ディスクつきとあって，値段は£3.95，日本では2,000円から3,000円と少し高め，当然両者とも内容はAMIGA一色，ディスクもAMIGAでしか使えないので，ユーザー以外の人が毎月購読するのはちょっと苦しいだろう。しかし，AMIGAから移植されるゲームもまだまだ多いし，AMIGAの周辺機器にはユーザー以外に興味深い（かつ，うさんくさい？）ものが多数ある。一度くらい買ってみて，そのヘンなパワーを体感しても損はしないと思う。

ところで，両者ともにPDS紹介のコーナーはずいぶん前からあったけれど，CU AMIGAではそれに点数をつけて掲載している。これはちょっと失礼ではなかろうか。

AMIGA WORLD

ヨーロッパのAMIGA専門誌を紹介したついでにアメリカのAMIGA専門誌を紹介して，欧米の雑誌を比べてみたい。

アメリカの雑誌はケバケバしいヨーロッパの雑誌に比べると，ずいぶん上品である。このAMIGA WORLDもすっきりとしたレイアウトで，スマートだという印象を受ける（本が本当に薄いということもあるが）。

写真点数は少ないながらも，図やイラストが多く，ある意味では雑誌の王道を行っている。しかし，パワーは感じられない。

内容はソフトレビューがただパラパラと並んでいる感じだし，日本では雑誌に3.5インチディスクをつけられないが，アメリカでも同じような決まりがあるのだろうか，ディスクつきの雑誌も見当たらなく，それを目当てに買うということもない。ユーザー以外の方々には輪をかけてつまらないだろうから，買って読んでみてくださいとはとても勧められない。

成熟しているということはつまらないことである，というような文句が頭に浮かんでしまうのはなぜだろう？

ただ，アメリカの雑誌であるから値段はドルで書かれていて，多少わかりやすい。

そのほか

ほかにももちろんMacintoshやWindowsの専門誌があるが，読者の皆さんが読んであまり面白いものではないだろう。

広告を見て周辺機器などを個人輸入する，というのであれば，それらの本はとても役に立つかもしれない。

グラフィック関係はどうなのだろうかと，「COMPUTER GRAPHICS WORLD」と「CADENCE」という雑誌も購入してみたが，ほとんど高価なCADの話ばかりでイマイチであった。

というわけで，そろそろ終わりにしたいと思うのだが，少しつけ加えておこう。

海外の雑誌すべてに共通していえるのは，歯に衣を着せないということである。

つまらないところをバシッと指摘し，評価も100点満点の15点とつけたりする。西洋人の考え方は昔からそうなのだとはわかっていても衝撃，そして刺激を受ける。

少し前の話になるが，「Days of Thunder」というレーシングゲームに対するある雑誌の評価記事では，以下のようなひとことで締めくくられていた。

“（このゲームを買わずに）これから出る「Indy500」というゲームを買うために，お金を使わないことを勧める”

＊　＊　＊

不正確な，あるいは少なすぎる情報は，時に素晴らしいアイデアの源となる。「よくわからないが，このソフトはこんなふうなのではないだろうか？」「きっとこういう機能がついているに違いない」。そんなことから新しいシステム，機能などが発想されることもある。そういう意味でも海外に目を向けることはよいことなのでないかと思う。

ながらく愛用していたパソコンが故障してしまった。と，書けば「んじゃ修理すればいいでしょ」といわれる程度のことだろうが，コトがハードディスクとあっては，ダメージたるや，ハンパではない。

なにしろ，マシンが年代モノ，アンティークに近いPC-9801VM 2。しかも容量が20Mバイトの外付けハードディスク装置で，インタフェイスもSASI，ときているから，見てもらったショップの兄ちゃんも早々にギブアップ。

店員：SASIのインタフェイスの古いハードディスクって，最近のマシンでは同期とれなくて，テストしてあげられないんですよねぇ……。

筆者：98なんかどれだって似たよーなもんでしょうが。(そこにあるパソコンを指して)これなんか，VMとさして変わらんでしょーが。

店員：あのー，これ486マシンなんですよねえ。

思わず「うわあっ」と叫んでしまう。

みじめな気分を通り越して，浦島太郎になったような気がして爽快ですらある。いつのまにかパソコンの世界も様変わりしてしまっていたようだ。

しかし，爽快さを味わって楽しんでいる余裕は，実際にはない。なにしろ，20Mバイトしかないゆえに，ありとあらゆる貴重なデータがつまっているハードディスクである。

書きかけの極秘文書も貴重だが，やはり過去に蓄積した大量の山が最大級の問題なのである。各種統計資料やら取材先企業のここ数年の財務データなどLotus1-2-3の表データの山，ニューススクラップやら2年分の映画リストやらの膨大なデータベース群，そしていうまでもなく，さまざまなテキストファイル。ぼくの場合はパソコン通信のとあるコーナーの編集長(そのネットではSYSOPといったりする)をやっていたりするから，その関連の記録も超大量に保存してあった。

「そーいうこともあるから，ちゃんとバックアップ取るようにという慣習がコンピュータ業界にはあるんですよ」

なんて，したり顔して説教する人もいる。

しかし待て。

本当にハードディスクをバックアップしている人が多いのであろうか？　少なくとも，ぼくのまわりでバックアップしているのは，ノートパソコンで少しのデータしかハンドリングしていない人ばかりだ。それもほとんど転送のついで，という程度。

「100Mバイトのハードディスクを買っちゃったよ」

なんて自慢する人が，バックアップしているケースは見たことがないし，なにしろ40Mバイトより容量が多くなると，バックアップする光景だけでもおぞましいものがある。100枚のフロッピーディスクを机に積み上げて，差しては複写，差しては複写，と作業が続いていくのだろうから。

ストリームテープデッキがどの程度売れているのかは知らないが，少なくともハー

X-OVER・NIGHT

(クロスオーバーナイト)

[第28話]

バックアップ

Oh!PC11/X-OVER-NIGHT

TAKAHARA HIDEKI 高原 秀己

ドディスクの1割も売れているということはあり得まい。ということは，いきおい，大多数の人がバックアップはしていないし，したくても量的にとても追いつかない。

とはいえ，現実にこういうことが起こってしまっては，やはりバックアップは必要，という結論になる。少なくとも，無事に買い替える日を迎える人ばかりではないはず。いくらかの人たちはXデーを待ちながら日日パソコンを使い続けていることになるわけで，考えてみれば非常に恐ろしい話である。

「自分だけはAIDSにはかからないだろう」という話と同じだろうが，日本での確率からいうと，AIDSにかかるよりは，ハードディスクトラブルに巻き込まれる確率のほうがまだまだ遥かに高いであろう。もちろん「うちの娘に限って」の確率よりは低いかもしれないが。

さて，それではどうすれば自衛できるのだろうか？　わざわざストリームテープデッキを買って，週1回バックアップを取るなどという作業は想像するだけで気が遠くなってしまうし，なによりも，そういう性質の機械は購入しないであろう。

では？

決して統計的な話でもないし，識者の声をまとめた話でもないので，故障して開き直っている人の戯れ言だと思って読み流していただければ幸いなのだが，バックアップを手軽に取る方法は，ズバリ，二重化システムが最適なのではないだろうか？

すなわちハードディスクを常に2台使用することにする。といっても，同時に2台購入するのは資金的にもキツい。だからとりあえず1台買って，どんな機械でも壊れる可能性が高まるといわれる3年経過の時点で，やや容量の大きい2台目を下落した価格で購入する，という方法。

3年もたつと，メモリチップ同様にハードディスクの世界では，相当なデフレーションが生じているので，購入はかなり楽になる。つまり，今100Mバイトのディスクを15万円で購入したならば，3年後には,200Mバイトディスクも10万円程度で購入できることになっているはずだ。その程度の価格ならば，そう辛くはないし，なによりも買い替えだと思えば，無理がない。で，使用するのは容量が大きいほうにして，小さいほうをバックアップ用とする。

当然ながら，フロッピーディスクも元媒体として使用しているのであるから，プログラム関係などすべてがバックアップ対象になるわけではない。容量が小さなほうをバックアップ用とする。ここで肝心なのは，常時双方を接続しておくことで，バックアップの繁雑さは確実に微小となる。

で，また3年たったら，大きい容量のものを買い増していく方法をとる。このようにすると，常に時代の先端を走れるし，バックアップもしている優良ユーザー，ということで自慢できる。

ちなみにぼくのハードディスクはメーカーへ修理に出してある。直るかどうかはわからないそうだ。

料金受取人払

高輪局承認

1396

差出有効期間
平成6年7月
15日まで

郵便はがき

1 0 8 - 0 0

5 0 7

(受取人)

東京都港区高輪
2-19-13 NS高輪ビル

ソフトバンク株式会社

Oh!X 編集部行

□□□-□□

電話

住所

氏名

年齢

職業・勤務先
学校・学部・学年

●編集部へのメッセージ

今月号の特集について

いちばん良かった記事	興味のなかった記事
これから載せてほしい記事内容	本誌以外にお読みのパソコン雑誌

推薦する市販ソフト

ソフト名：

推薦理由：

パソコンにどんなオモチャをつなげたらいいと思いますか？

あなたの愛機は（所有機種に○印をつけてください）　ない

X1（マニアタイプ,C,D,F,G,twin）X1 turbo（model 10,20,30,40,II,III,Z,ZII,ZIII）

MZ-（80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861）

X68000（初代,ACE,PRO,PROII,EXPERT,EXPERTII,SUPER,XVI,Compact, HD）

その他　MIDI楽器（　）

FD（　基）　TAPE　QD　HD（　MB）　MO　プリンタ（　）

年齢　歳　パソコン歴　年　男・女　プレゼントNo.

通常払込料金
加入者負担

払込通知票

口座番号	東京	1	−	29307	金額	
加入者名	ソフトバンク株式会社				料金	払込み / 特殊 円
払込人住所氏名	※ （郵便番号 ）				備考	
					受付局日付印	

各票の※印欄は、払込人において記載してください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。 （郵政省）

切り取らないで郵便局にお出しください。

通常払込料金
加入者負担

払込票

口座番号	東京	1	−	29307
加入者名	ソフトバンク株式会社			
金額				
払込人住所氏名	※			
備考		受付局日付印		

振替用紙

←点線から、きれいに切り取ってご使用ねがいます。

【定期購読のご案内】

●定期購読のお申し込みは、この郵便振替用紙のみとさせて頂き、銀行振込・現金書留によるご入金は、ご遠慮下さい。

●受付締切は、

1日発売　：　発売日前月10日振込
8日発売　：　〃　15日振込
15・18日発売：　〃　25日振込
です。

〈例〉 4月1日発売（Oh！PC 4月15日号）の場合、お振込の締切は3月10日です。
締切に間に合わなかった場合は、自動的に次号からの発送となります。
なお、すでに発売されているもの、また、お振込が締切に間に合わなかった月号のものは、定期購読ではお求めになれません。書店でご購入ください。

●定期購読誌のお届けは書店発売日より遅くなりますのでご了承下さい。

「発売日一覧」

◇毎月1・15日発売
Oh！PC

◇毎月18日発売
Oh！X
Oh！FM TOWNS
C MAGAZINE
Oh！Dyna

◇毎月8日発売
月刊情報処理試験
LANTIMES
THE WINDOWS
DOS／Vmagazine
UNIX USER

切り取り線

切り取らないで郵便局にお出しください。

この欄は、加入者あての通信にお使いください。

送り先

お名前	フリガナ	お電話	
ご住所	フリガナ 〒		

定期購読申込書

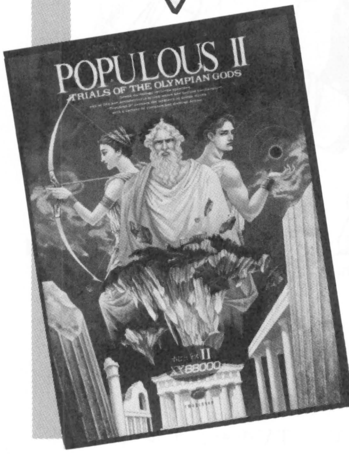
□Oh！PC	年間	（23回）	12,880円	（新規／継続 NO.　）
□Oh！PC	6ヶ月	（12回）	6,720円	（新規／継続 NO.　）
□Oh！X	年間	（12回）	7,200円	（新規／継続 NO.　）
□Oh！FM TOWNS	年間	（12回）	7,440円	（新規／継続 NO.　）
□C MAGAZINE	年間	（12回）	11,760円	（新規／継続 NO.　）
□Oh！Dyna	年間	（12回）	9,120円	（新規／継続 NO.　）
□月刊情報処理試験	年間	（12回）	8,160円	（新規／継続 NO.　）
□月刊情報処理試験	6ヶ月	（ 6回）	4,080円	（新規／継続 NO.　）
□LANTIMES	年間	（12回）	17,760円	（新規／継続 NO.　）
□THE WINDOWS	年間	（12回）	11,760円	（新規／継続 NO.　）
□DOS／Vmagazine	年間	（12回）	9,360円	（新規／継続 NO.　）
□UNIX USER	年間	（12回）	11,760円	（新規／継続 NO.　）

〔備考欄〕

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。　（郵政省）

イマジニア ☎03(3343)8911

1 ポピュラスII



X68000用
5"2HD版2枚組

12,800円(税別) 3名

このゲームはAMIGA版が発表されたときに国内でも話題になったけれど，移植一番乗りということでX68000ユーザーを喜ばせてくれた。さらにリアルになった神々の戦いを堪能しよう。

エニックス ☎03(3371)5581

2 ドラゴンクエストV

スーパーファミコン用

9,600円(税別) 1名

少し前に世間で話題のタネになっていた「ドラゴンクエストV」をエニックスの関係筋からいただきました。X68000とは全然関係ないんだけど，縁起モノとしてどうぞ。

シャープ ☎03(3260)1161

3 SX-WINDOWイラスト集 Vol.1&2

X68000用 3.5/5"2HD版

各8,000円(税別) 3名

PIX形式のイラスト集。Vol.1の一般実用編はスポーツ，ビジネス，食べ物など，Vol.2の行事・四季編は季節の行事，風物詩が収められているのでうまく利用しよう。

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1992年11月18日の到着分までとします。当選者の発表は1993年1月号で行います。

ソフトバンク ☎03(5488)1360

4 追補版 SX-WINDOW プログラミング

4,200円(税込) 5名

この本が出たのはSX-WINDOW ver.2.0が発表される前で，ver.1.10対応版になっているけど，基本的な部分は共通だからお役に立つこと間違いなし。

キングレコード ☎03(3945)2111

5 Sound Wind ミニCD

非売品 10名

「イース」の女神フィーナの歌声をイメージして作られたCD「Sound Wind」が発売されているんだけど，その販促用ミニCDをプレゼント。歌っているのは南翔子さんです。

9月号プレゼント当選者

1 グラディウスII （京都府)天達雄一 （福岡県)伊藤竹虎 （大分県)森脇弘之 2 ヘビーノヴァ （栃木県)柿沢雄大 （東京都)足立義宗 川勝博剛 （神奈川県)相馬信隆 （奈良県)前田武志 3 電子手帳用フラッピーカード （北海道)佐々木淳一 （兵庫県)赤城豊和 4 テレホンカード各種 （千葉県)星野こずえ （東京都)森嵜美千代 （愛知県)鈴木達也 （広島県)北中詠司 5 上昇気流 vol.3 （宮城県)千葉浩貴 及川雄也 （千葉県)浅野一行 （長野県)荒井俊矢 （埼玉県)福田幸彦 （東京都)田中義彦 （静岡県)佐藤剛 （大阪府)肥後有紀子 （香川県)早野哲也 （宮崎県)山口宏幸 （敬称略）
以上の方々が当選しました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，雑誌公正競争規約の定めにより，当選された方はこの号の他の懸賞には当選できない場合がありますのでご了承ください。

(で)のショートプロぱーてぃ——その38

文明の窓・テレビであるのだ

Komura Satoshi 古村 聡

illustration : T. Takahashi

諸国漫遊の旅を終え，やっと(で)氏が帰ってきました。ますます元気に，ますますC調になったショートプロが始まります。今月は，作業中にテレビが見れる「NTV.C」とグラフ作成ツール「EINSTEIN.BAS」の2本です。

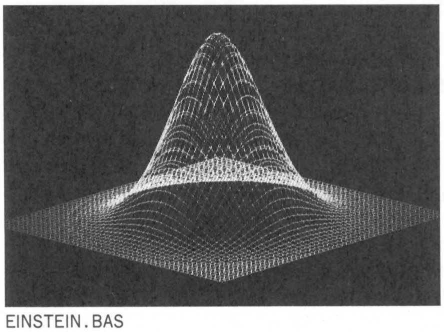
どもー，ただいま帰ってきました(で)でございます。え，お前は誰だって？

私を忘れるとはなにごとだっ！　私こそが正真正銘真性B型刻苦勉励驚天動地……つまり，私こそが本当のこのコーナーの主，(で)様なのである，えーい，控えおろう！

はぁはぁ。ちょっと旅に出ると，すぐこれだからなー。

ま，なにはともあれ，休んでいる間代理をつとめてくれた影山&大和君，どうもありがとさんでした。ついでに「(で)よ，君のことは忘れない」などというリードを書いてくれた編集担当さんも……。本当に忘れてないんでしょうね，まったく。ま，しかし，こっちも半年間原稿書かなかったわけだから，ちゃんと書けるかどうかわからなかったりするわけでお互いさまだったりするし……いいんですけどね。

それはともかく，この旅に出ている間に私が忘れ去られてしまったように(オイオイ)，私にもこの半年間で，原稿の書き方以外にもすっかり忘れてしまったものがあるのです。実をいうと私，この半年間というもの文化的な生活をしていなかったために，まったくテレビを見ていなかったのですよ。ああ，おかげさまで新番組なんか全然知らないしー。オリンピックがあったのも知らなかったしー。岩崎恭子ちゃんなんか出た日には「アイドルじゃなさそうだし……誰だこれ？」てなもんですもん。わーっ，あっという間に浦島太郎や。でもこれは冗談抜きで，しばらく必死でテレビを見ないと社会復帰できないような気がしますね。うーむ，おそるべし情報化社会日本……。



EINSTEIN.BAS

そうそう，復活祝いというわけでもないけど，今月から「動かないよ，と思う前に」という小さなコーナーを設けてみました。内容は見てのとおり，ごくごく基本的なことだけれども知っていないとどうにもならない。そんなことを中心に紹介していくつもりです。

X68000だからこそテレビなのだ

ではでは，プログラムを紹介しましょうか。1本目は兵庫県の秋月さんによるプログラムでNTV.Cです。

NTV.C
(要ディスプレイテレビ，XC Ver2.0以上)
兵庫県　秋月　誠

んー，これはなにをするものかというと，「作業中テレビを見るためのプログラム」なのです（リスト1）。たとえばCで書かれたプログラムをコンパイルしている間，非常に暇ですよね。そんなときに，作業をしている間だけ，ディスプレイをテレビに切り替えてくれるという，非常に便利なものなのです。

このプログラムはC言語で書かれていますので，XC ver.2.0以上のコンパイラを使ってコンパイルしてください(GCCも使えます)。

XCなら，

```
A>cc /Y NTV.C
```

GCCなら，

```
A>gcc -lfloatfnc -liocs NTV.C
```

でOKです。

で，使い方ですが，

```
A>NTV -TVチャンネル コマンド
```

としてやります。たとえば，XCでTEST.Cというプログラムをコンパイルする間，8チャンネルを見たいのならば，

```
A>NTV -8 cc test.c
```

とするだけ。簡単でしょ。これで，テレビにディスプレイが切り替わり，コンパイルが終わると同時にコンピュータのディスプレイに切り替わります。あ，チャンネルは1～12までの間なので注意すること。

ちなみに一度にいろいろなことをしてしまいたいときには，

```
A>NTV -12 cc test.c ‖ copy test.x b:
```

のように“‖”で区切れば左から順に実行してくれます。

当然のようにテレビの映らないディスプレイじゃ使えないし，テレビコントロールケーブルがきちんとX68000に差さっていないとディスプレイが切り替わらないので注意してくださいね。そんな人はいないとは思うけど(私ゃ，やってしまった……，違うっ，あれはしばらく留守にしてたから誰かが抜いてしまっていたんだ！　ううっ)。

いままではコンパイル中にテレビを見ようとすると，何度も何度も画面を切り替えて作業が終わるのを気にしたり，下手するとテレビに夢中になってすっかりプログラミング中であることすら忘れてしまったりしていたわけですけど，これでもう時間を有効に使えるってもんですね。もっとも今度はテレビのほうが見たい場面だったのに，コンパイルが終わってしまってプチっ，ていう可能性もあるわけだけど。

あ，そうだ。このプログラムは少し手を加えさせてもらいました(その作業中に私もこのプログラム，たっぷり使わせていただいたりしたのだな，これが)。

んー，やっぱりテレビはいいですね。なんだか，原稿書きながら，テレビを見て，プログラム書きながらテレビを見て，とやっていたらすっかり原稿も書けてしまいましたし。テレビ番組を見ていると，文明っていいな，という実感がわいてきましたよん。そうだ，せっかくですから，満喫した情報のその成果をここで皆さんにお見せしましょう。せーの！

「美少女戦士セーラームーンはいいぞ，ついでに亜美ちゃんは俺のもんだ」(な，なにを見てたんだいったい)

ふっふっふ，こりゃ社会復帰できる日も近いな。

気分はE=MC²

さーて，ではでは今月の２本目いきましょう。福井県の大久保さんによる「アインシュタインロマン」であります。どーぞ！

EINSTEIN.BAS for X68000

(X-BASIC)

福井県　大久保敏之

このプログラムは，z=f(x, y)で表される曲平面を表示するプログラムです。BASICプログラムですので，リスト２を打ち込みRUNするだけで，ライン描画による式の3Dグラフィックが表示されます。もちろん，式は自在に変えることもできます。

おお，カッコいいではないですか。こういうことが結構簡単にできてしまうんだから，BASICもまだまだたいしたもんだと思ってしまいますね。こいつぁ復帰早々縁起がいいぜ。

グラフの形を変えるには，プログラム中の400行にグラフとして描画される式が書かれていますので，これを書き換えることでいろいろな曲面が表示できます。ただし曲面が透けて見えるため，多少は試行錯誤が必要ですけど(式によっては汚くなってしまう)。まあ，さすがにこれだけリストが短いと陰面処理はむずかしいですからね(片方の軸からだけならペイントでごまかすっていう手もあるけど……。このプログラムは両軸使ってるからその手も使えないしね)。

あ，どうしてもグラフの形を変えるのがうまくいかないっていう人は，リストの410

動かないよ，と思う前に(1)

なんだか，家電の説明書の最後のページのようですけど，実はそのとおり。Ｃやアセンブラで書かれたソースリストが掲載されるとよく，「動かないんですけど」と編集室にいわゆるバグ電話してくる方がいらっしゃいます。でも，これにかかってくる電話って，打ち込まれた方がミスしていることが多いのが現状なんですよね。わりとありがちなミスを挙げておきますので，電話する前に一度確認してみてくださいね(本当にバグのあった場合にはバグ電話に報告してください)。今回はＣコンパイラで使う環境変数などについてです。

★PATHは通ってますか？

AUTOEXEC.BATに書かれることが多いPATHですが，CC.XやLK.XのあるディレクトリをPATHで指定していますか？ この環境変数はコマンドラインで“PATH”と打つことで確認することができます。このミスの場合は「コマンドまたはファイル名が見つかりません」というメッセージが出るので，わかると思いますけどね。

★環境変数は設定していますか？

“SET”と打ち込んでみましょう。

lib=……

include=……

SET環境変数が設定してありますか？ これらの環境変数がないとＣコンパイラは正しく動いてくれません。“LIB=”がなかった場合，「～.a(.l)が見つかりません」といってきます。また，includeで指定されたファイルがなかった場合，プリプロセッサの段階で「#include命令のファイルが見つかりません」というエラーになります。環境変数もAUTOEXEC.BATに書かれていますから確認しましょう。

★オプションスイッチは正しいですか？

XC Ver2.0以降では関係ありませんが，古いコンパイラではオプションスイッチの大文字，小文字が違っていると正しくコンパイルしてくれません。たとえば，

A>cc /Y /W test.cを，

A>cc /y /w test.c

と間違えてしまうとリンクの段階で，

unfinded symbol(s)

というエラーメッセージが出てきてしまいます。コンパイラのエラーでなくリンカでエラーが出ますので見つけにくいようです。コンパイルするときには注意しましょう。

リスト1　NTV.C

```
/**********************************************************
        にこにこTV    Ver 1.00
        ^_^  ^_^
                                        1992,8,19
                              By Akky
    改変   by (で) 1992.09/17
**********************************************************/
#include        <stdio.h>
#include        <stdlib.h>
#include        <iocslib.h>
#include    <string.h>

/*プロトタイプ宣言*/
void    error(int);

void    main(int argc,char *argv[])
{
        int     i,k;
        char    command[1024];/*実行するコマンドが入る*/

        /*変数の初期化*/
        i=1;

        /*コマンド文字列がわたされなかった*/
        if(argc==1)
        {
          printf("使用法:NTV  [-TVチャンネル] [コマンド]¥n");
          exit(0);
        }

        TVCTRL(0x1c);    /* テレビ画面にする */

        /* '-' または '/' でチャンネルが指定されたとき*/
        if(argv[1][0]=='-' || argv[1][0]=='/'){
        k = atoi(&argv[1][1]);
                if((k<=0)||(k>=13))
                        error(0);
                TVCTRL(0x0f+k); /* チャンネルをセットする */
                i++;
                if(argc==2){
                    TVCTRL(0x1d); /* コンピュータ画面にする */
                    exit(0);
                }
        }

        /*文字配列commandにコマンド文字列をセット*/
        command[0]='¥0';
        for(;i<argc;i++){
                        strcat(command,&argv[i][0]);
                        strcat(command," ");
        }

        if(system(command)<0)
                error(1);
        TVCTRL(0x1d);    /* コンピュータ画面にする */
        exit(0);
}
void    error(int i)
{
        if(i)
        {
                printf("エラーが発生しました¥n¥a");
        } else {
                printf("チャンネルの指定が異常です¥n¥a");
        }
        TVCTRL(0x1d);    /* コンピュータ画面にする */
        exit(-1);
}
```

～490行におすすめのサンプル式が一緒に載せてありますので，そちらを打ち込んでみるなり参考にするなりしてみてください。このサンプルは結構きれいですよ(410～490行の式を入れるときには400行を削るのを忘れないようにね)。

ところで，作者の大久保さんの投稿原稿によると，このプログラムのタイトルは，NHKの「アインシュタインロマン」でやっていた重力による空間の歪みを表現したグラフィックにヒントを得たからなのだそうです。へー，アインシュタインですか。するってーと，フジテレビ系列でやっていて，AMIGAのグラフィックを使っていて，アナウンサーの城ケ崎さんと松尾さんが出てきて，Winkみたいに踊ってしまう科学番組を思い出してしまうけど。

というわけでまた来月。んじゃね。

リスト2　EINSTEIN.BAS

```
 10 /*
 20 /*歪んだ平面(曲面)を表示するプログラムVer1.30
 30 /*
 40 screen 2,0,1,1
 50 int x,y,x1,y1,x2,y2 : float x3,y3,z
 60 /*line(-17,255,383,155,15)
 70 /*line(783,255,383,155,15)
 80 /*line(783,255,383,355,15)
 90 /*line(-17,255,383,355,15)
100 for x=-25 to 25
110   xyz(x,-25)
120   x1=383-x*8-25*8
130   y1=255+x*2-25*2-z*30
140   for y=-49 to 50
150     y3=y
160     xyz(x,y3/2)
170     x2=383-x*8+y*4
180     y2=255+x*2+y*1-z*30
190     line(x1,y1,x2,y2,14)
200     x1=x2
210     y1=y2
220   next
230 next
240 for y=-25 to 25
250   xyz(-25,y)
260   x1=383+25*8+y*8
270   y1=255-25*2+y*2-z*30
280   for x=-49 to 50
290     x3=x
300     xyz(x3/2,y)
310     x2=383-x*4+y*8
320     y2=255+x*1+y*2-z*30
330     line(x1,y1,x2,y2,14)
340     x1=x2
350     y1=y2
360   next
370 next
380 if inkey$="" then 380 else end
390 func xyz(x;float,y;float)
400   z=10*exp(-x*x/100)*exp(-y*y/100)-3
410 /*z=(-x*x-y*y)/100+5
420 /*z=cos(sqr(x*x+y*y)/4)*2
430 /*z=(-sqr(x*x+y*y)+10)/2
440 /*z=-10*exp(-(x-4)*(x-4)/100-y*y/100)+10*exp(-(x+4)*(x+4)/
100-y*y/100)
450 /*z=pow((x*x+y*y),2)/360000-5
460 /*z=-4*exp(-x*x/100)*exp(-y*y/100)+8*exp(-x*x/16)*exp(-y*y
/16)
470 /*z=cos(sqr(x*x+y*y)/2+cos(sqr(x*x+y*y)/2))
480 /*z=-10*exp(-(x-2)*(x-2)/9-y*y/9)+5*exp(-(x+4)*(x+4)/100-y
*y/100)
490 /*if x=0 and y=0 then z=-1000 else z=-1#/(x*x/100+y*y/100)
500 /*↑↑↑↑↑↑↑↑↑↑↑↑↑↑
510 /*0で割ることになってしまうのでごまかしています。
520 endfunc
```

ぱーてぃハンズ(1)

(で)のスーダラプログラム術

えー，めでたく新装開店いたしましたぱーてぃハンズでございます。や，めでたい，めでたい(単にページの飾りが変わっただけという話もあるけど……)。

それはそうと，いま，私がいなかった間のぱーてぃハンズを読んでいるんですが，すごいですね。今月，いつかやったゲーセン対戦記みたいなものでも書こうかと思っていたんですけど，影山氏にかわってもらったおかげですっかりまじめなプログラム講座になってしまってるんだもんな，このコーナー(おいおい，本来ここはマジメなプログラムのコーナーだぞ)。やっぱり影山氏マジメだから。性格って原稿にも出るんですね。

しかし，ずぼらなやつにはずぼらなりのプログラムの組み方だってあるんです。いいかげんに軽～く組むプログラムだって結構楽しいんだし。

ということでしばらく本プログラム講座ぱーてぃハンズは「スーダラ(で)の気楽なプログラミング講座」というセンでいってしまいます。では，新しくなったぱーてぃハンズ，第1回目いきます！

お気楽だからBASIC

さて，お気楽プログラミングでありますから，ここではおもむろに，

　A>basic

としてBASICを立ち上げます。世の中でいちばんお気楽という言葉が似合うプログラミング言語，それはBASIC。いまさらこんなことをいってしまうほど，BASICってのは気楽に使えるいいプログラミング言語なのです。だって，BASICはタダで本体についてくるでしょ。それから，作ったらとりあえずRUNしちゃえばいいでしょ。暴走しないでしょ。X68000のX-BASICでは，それまでCやPASCALでしか使えなかった「関数」というものも使える。X68000にはまだないけど，最近のプログラム界のトレンド，オブジェクト指向のBASICまでありますからね。それほどまでにBASICは王道なのであります。

んでなにを作りましょうか。シューティングは前にやってますね。迷路もやったし……そういえば，最近のトレンド(というかもうほとんどこれしかないとゆー気もするのだが)なゲームのジャンルがひとつありますよね。

そう，あえて，ゲーム名は挙げませんけど(笑)，最近はゲームといえば対戦型大キャラ格闘技ゲーム，略して「格ゲー」ですよね。こいつがいいな。ってことは，まず画面があって，大きな登場人物，キャラクタがいて，で，こいつを2人の人間がジョイスティックで操る。んでもって，ボタンで殴ったり蹴ったりして，力尽きたほうが負けと。キャラクタはスプライトで作ればいいかな。うん，決まり！　今回は格闘ゲームでいきましょう。

ゲームの命スプライト

で，今回はスプライトをミソとして使うわけです。スプライトってなんだっけ，というX68000ユーザーは(たぶん)あんまりいないんじゃないかな，という気もするんですけど，軽く説明しておきましょう。

ずばり，スプライトとは「画面上に，ある小さなパターンを描くもの」のことなのです。

普通パソコン上でグラフィックで絵を描く，というと，これはそのままグラフィックの画面全体が大きな紙のようになっていて，その上に絵を描いていくわけです。ところが，スプライトを使って絵を表示する，ということは，タテヨコ16×16ドットか8×8ドットという小さな紙のようなものに絵を描き，そして絵を描いた小さな紙を大きなグラフィックの紙の上にのせ

る，ということをして絵をディスプレイ上に描いているのです(図 1)。

なぜ，こんなに面倒なことをしているか説明しましょう。たとえばゲームを作っているとしますね。そのときに「キャラクタを素早く，スイスイっと動かしたい」ということが要求されたとします。

このときにグラフィック(大きな紙)に直に絵を描いてしまうと，描いた絵をまずいったん消して，それからもう一度同じものをずらして描かないと動かせないのです。同じものをもう一度描き直すわけですから，当然，時間がかかってしまいます。

しかし，動かしたいキャラクタをスプライトに描いておけば，スプライト(小さな紙)をほいっと持って動かすことができるわけです。

てことは，スプライトのほうがゲームなどのキャラクタを表示したり，消したり，移動したりっていうのが楽にできるんだ，というのはわかりますよね。わかったかな？

ああ，どうしよう大きなキャラ

ところでところで。ふと，私は悩んでしまいました。というのも，さっきもいったようにスプライトというのは小さな絵を表示するためのものです。正確にいうと，スプライトというのは，1 個で最大16×16ドットしかないんです。今回のようにデカキャラゲームのキャラクタは明らかにその5〜6倍の大きさがいるんですよね。16×16ドットの大きさじゃ，画面がいちばん粗いモードでもRPGのチビキャラぐらいの大きさしかないわけで……，チビキャラの格闘ゲームってのも一興かもしれないけど。

「いいや，スプライト 6 個分の大きさなんだから 6 個並べちゃえ」

というわけで図 2 のようにひとつのキャラクタを 6 枚のスプライトに分けて描くことにしました。考えた末に，いちばん安易な方針でいっちゃうんだもん。あとでどうなるかわからないけど，あと先考えずに気楽にいく。これが，お気楽プログラミングのコツです(おいおい)。

んでは，方針が固まったところでさっそくスプライトに描いておく絵を作ってしまいましょう。これには標準で付属するDEFSPTOOL.BASというスプライトエディタを使います。BASICを立ち上げて以下のように実行してください。

run "¥ETC¥DEFSPTOOL.BAS"

さて，DEFSPTOOLをrunすると妙なことに気がつきませんか？　そ，スプライトエディタで大きな，このゲームに使うようなキャラクタを描けそうもないのです。

ま，考えてみれば当たり前の話ですよね。スプライトっていうのは大きくても16×16ドットなわけですから，普通に考えればそれ以上大きな絵を描く必要はないわけです。

でも，困った。大きな絵が描きたい。しかたがない……あなたならどうする？

私はまたしても，なにも考えませんでした。せっかく立てた方針を無駄にするともったいないでしょ(おいおい)。

どのみち，スプライトを 6 枚組み合わせて使うわけですから，パターンを 1 枚つまり全体の1/6ずつ描いていきます。DEFSPTOOLの左側にスプライトが並べて表示されますので，そのときに 1 枚の絵になるように並べていき，最後に 6 枚がうまくつながるように調整しました。たとえば最初の絵はスプライトの番号で0，1，8，9，10，11番のパターンにするわけです。

……と10行ぐらいでさらっと書いてたけど，死にましたこの作業で。実際1/6ずつ描くと当然のようにずれます。で，なんとかつじつま合わせると今度はキャラクタのデッサンが狂って，格闘というよりは阿波踊りのようになってしまうんですよー。何度描き直したことやら。単にこのへんの説明が短いのは思い出したくなかったからだったりします。で，この原稿を書いている途中で知ったのですが，私がいなかった号の付録ディスクに，でっかいキャラを描けるスプライトエディタが発表されてるじゃないですか。しかもBASICへのコンバート方法まで質問箱に出てるし。皆さんはちゃーんと，こういういいツール使って，正しく徹底的にお気楽プログラミングしてくださいね。ううっ。

教訓 1 ：下手なお気楽，地獄の始まり
　　　　楽にやりたきゃツールを選べ

スプライトを表示するのだ

でもって，描いたスプライトのパターンを画面に表示するわけですね。やっとBASICでプログラムを組むわけです。組むっていってもたいしたことはないんですよね。スプライト描くのって簡単ですから。まず，

sp_init()

でスプライトを初期化します。それから，

sp_disp(1)
sp_on(0, 1)

で，スプライトを画面に表示する下準備をします。で，実際に描いたパターンを画面上に表示するには，sp_move()関数で出したい座標にパターンを置いてやるだけなんですよね。今回は 6 枚のスプライトで 1 個のキャラクタにするから，このsp_move()関数を 6 回使って 6 枚のスプライトを表示してやるだけです。簡単でしょ。

そうしてできたプログラムがリスト 1 (参考)です。スプライトの表示はPutSprite()という関数を作って 6 枚のスプライトを一度に表示してみるようにしました。そうそう，あと，SCREEN命令でスプライトが使える画面モードにするのも忘れないようにしないと，変なエラーに悩まされてしまいますからね(途中でブレイクしてリストを書き換えるときには，WIDTH64して見やすくしたほうがいいですね)。

で，ゲームになったらこいつらを動かしてやればいいわけです。なんとなくゲームができそうな気がしてきたでしょう。ふふふふ。

でもでも，キャラクタデータはどうするの？と素朴な疑問を持った方，ごめんなさい。一応，使うデータを描いたはいいのですが，今回紹介しようとしていたパターンデータのリストが，あまりにも巨大になってしまい掲載できなかったんです。とりあえず，図 3 のパターンナンバーに適当なキャラクタを自分で描いて試してくださいね。

では，来月はこのスプライトをジョイスティックを使って動けるようにしてみます。

図1　スプライトは画面上に置かれた小さな紙と考える

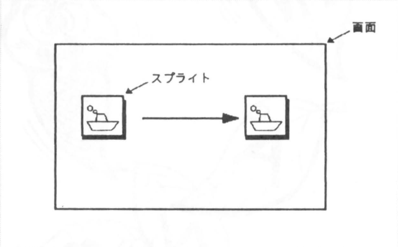


図2　ひとつのキャラクタを6枚のスプライトに分割する

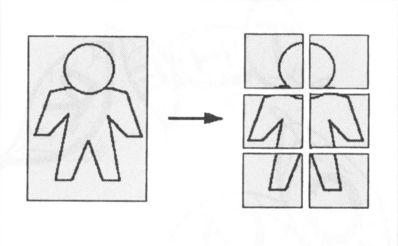
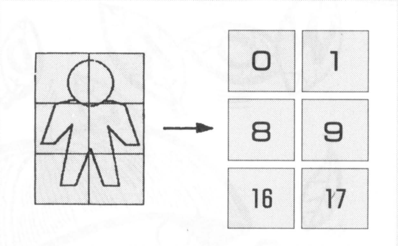
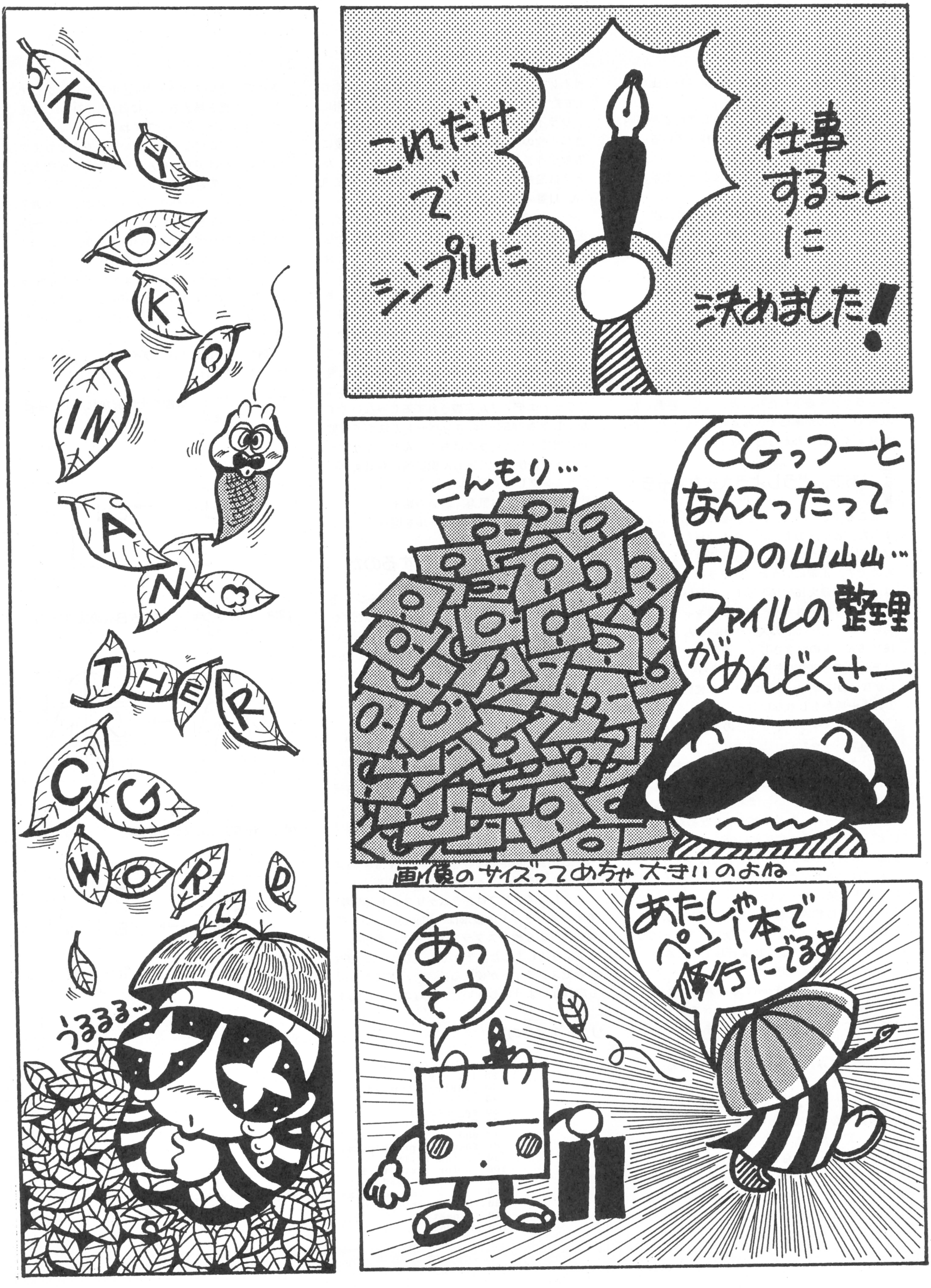


図3　対応するスプライト番号

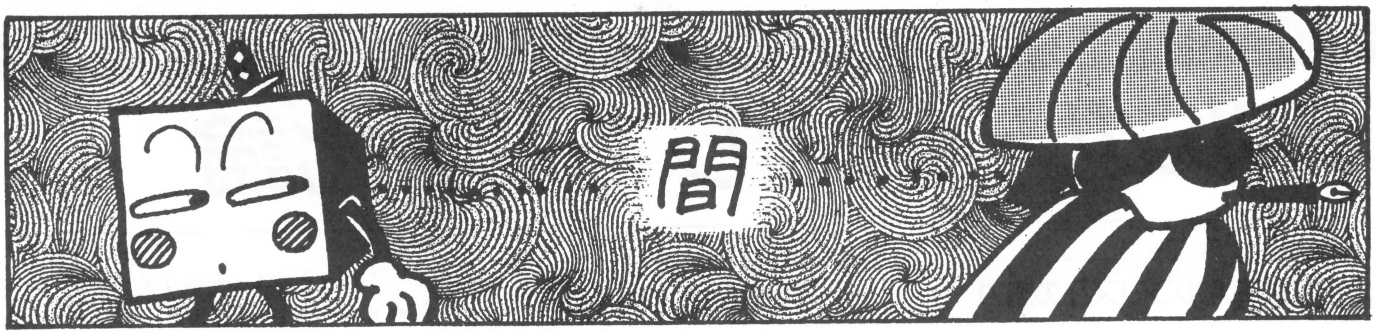


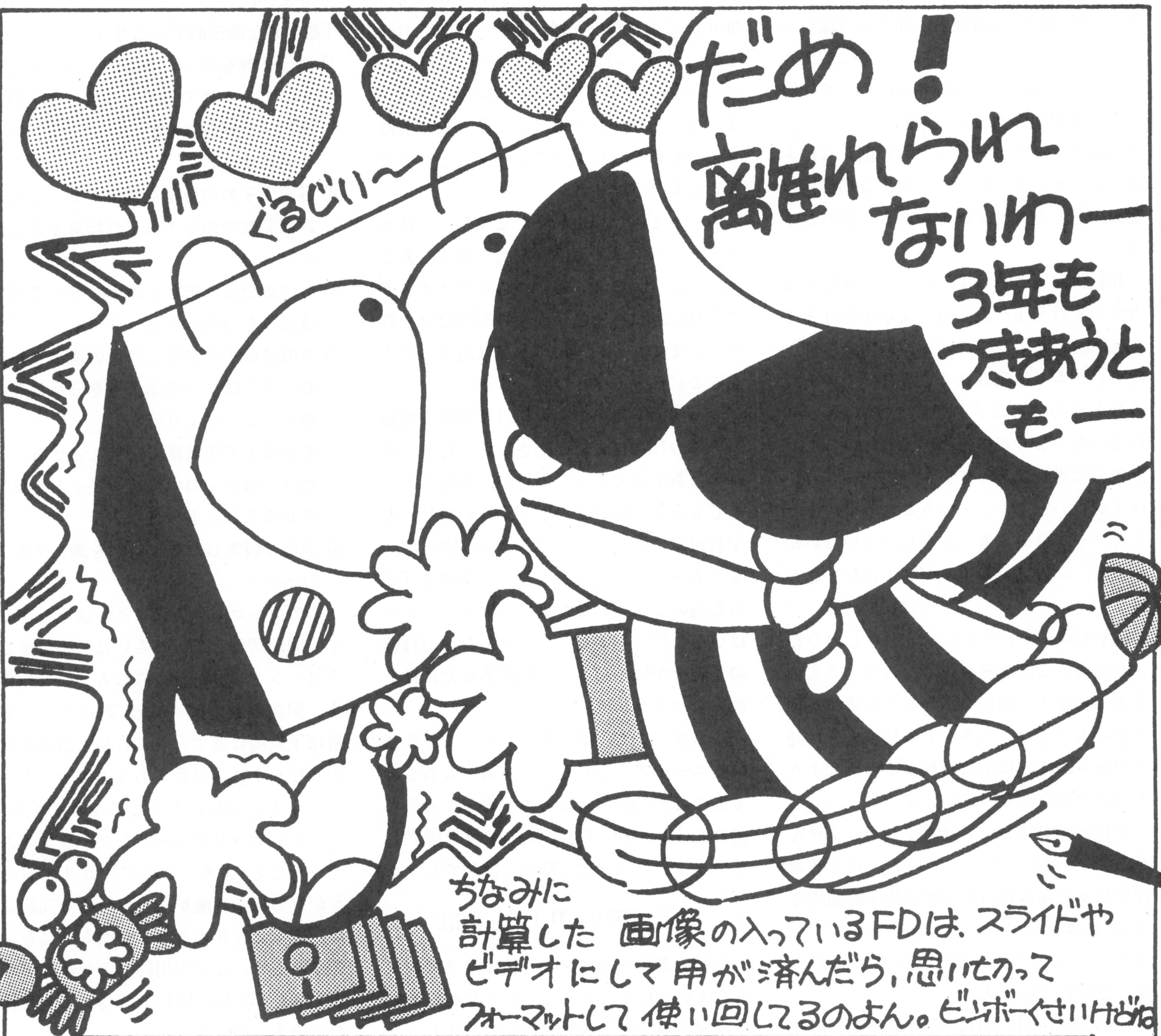
リスト1（参考）

```
 10 screen 0,0,0,0
 20 int x,y,h
 30 x=48:y=128:dh=-4
 40 sp_disp(1)
 50 sp_on(0,1)
 60 PutSprite(0,x,y)
 70 end
 80 func PutSprite(pnum, x, y)
 90    int ix,iy
100    for  ix=0 to 1
110       for iy=0 to 2
120          sp_move(ix+iy*8,x+ix*16,y+iy*16,pnum+ix+iy*8)
130       next
140    next
150 endfunc
```



これだけでシンプルに
仕事することに決めました！
こんもり…
CGっつーとなんてったってFDの山山山…ファイルの整理がめんどくさーー
画像のサイズってめちゃ大きいのよねー
あたしゃペン1本で修行にでるよ
あっそう
KYOKO IN ANOTHER CG WORLD
うるるる…





今回のCGデータ

総物体数 233(うちメタボール数32)
光源 2
1280×1024ピクセル
1670万色フルカラーを
4×5ポジで出力

使用ソフトは，
C-TRACE，サイクロン
マッピングデータ作成に
Z'sSTAFF PRO-68K

アルゴリズムとしての進化論

遺伝的アルゴリズムの魅力

どんな世界にも流行というものがありますが，情報科学の研究者の間で最近流行っている研究テーマのひとつに「遺伝的アルゴリズム」(Genetic Algorithm，略してGA)というものがあります。過熱してきたのはたぶん今よりも2〜3年ぐらい前からだと思いますが，今なおこれに対する期待感は衰えの兆しを見せていないように思われます。

比較的早くからGAに取り組まれた筑波大学の星野先生が書かれた文章があります(文献1)。そのサブタイトル「その信仰と現実」からも憶測されるように，この文章のひとつの主旨は「まだまだGAは解明されていない部分が多く，ブームにのってバラ色の世界が広がっていると思いすぎないほうがいいですよ」ということです。まだまだブームは衰える気配を見せていないからこそ，この文章の重みが増すのだと思われます。

GAはニュートラルネットをもしのぐひとつの大きな潮流を形作るだろうとさえいう人もいます。確かに，今ですら情報処理学会の大会でコンスタントに10本近くは発表があり，すでに地道な数多くの研究よりも目立つ存在になっています。

今後の動向についてはよくわかりませんが，GAの背景にひかえている，ある普遍性を持った説得力のある歴史的・時間的重み(それは進化論といってもいいのでしょうが)，そのようなものを持っている点に僕も惹かれています。

GAの大きな特長は，それが今までの技術にすぐに取って代わるような即戦力的な実用性を持っていると同時に，生命の成り立ちという壮大なテーマになんらかの手がかりを与えてくれそうだという期待感を与えてくれるところにあると思います。そして，このような特長を持つがゆえに，これからも，もっともっといろいろな領域の人人の関心を幅広く呼び起こしていくのだと思います。

Goldbergと禅問答

基本的な遺伝的アルゴリズム自体はもう30年ほど前にHollandによって提唱されました。最近このように盛り上がってきたのは，彼の後継者であるGoldbergの研究によるところが大きいのでしょう。彼は実際この分野の今日におけるリーダーといってよいでしょう。

ちょっと話は横道にそれますが，彼はユニークなスタイルの論文を書きます。ある論文では，論文の冒頭につけるアブストラクト(概要)に，次のような禅問答だけを載せています。もちろん，本当は論文の要約をここには書くべきなのですが。

——明治時代にある大学教授がある禅師のところに禅について聞きにいったところ，禅師は茶をふるまったのだが，茶碗にいつまでもそそぎ続けるのであふれ始めた。大学教授はこらえ切れずに，「いっぱいだ，もう入らない！」と言ったら，禅師は「この茶碗のようにあなたは自分の考えでいっぱいなのだ。どうして，まずあなたが自分の茶碗を空にしないで，私があなたに禅を教えられようか？」

これは，まあ，第一人者だからこそ許されるユーモアなのでしょう。第一人者である彼の著書(文献2)はもっかのところ最も優れたGAの教科書としての地位を確立しています。日本語化も時間の問題でしょう。

遺伝的アルゴリズムとは

生物の進化の過程をごく簡単なモデルにしたものがGAです。対象とする進化という現象の概念的な説明を最初にしましょう。まず，両親の遺伝子が組み合わされて子供の遺伝子が出来上がります(交叉)。その際には，遺伝子の一部が無作為に変化することがあります(突然変異)。そして，そのような生物群は自然淘汰，適者生存という時間というものに必然的に内包されたメカニズムの試練を受けることにより，結局，遺伝子レベルでみれば自然界に適した遺伝子が残っていくというわけです。

これをそのまま単純なアルゴリズムにしたものであるGAは次のようなステップから構成されています。

[基本的な遺伝的アルゴリズム]

1) 遺伝子群を作る。各遺伝子は通常0か1のビット列で表す。このビット列は適用する問題(評価関数)のパラメータを意味する。
2) 各遺伝子の適合度を評価する。これは，評価関数の値をそれぞれ計算することに等しい。
3) その適合度に応じた確率を与えて親の遺伝子を選択しコピーを作る。
4) その遺伝子を交叉し子供の遺伝子を作る。交叉は2つの遺伝子を分断し入れ替えることにより行う。
5) ある確率で突然変異を起こさせる。これは，遺伝子の中にあるビットを反転させることにより行う。
6) 適当に設定した終了条件を満たすまで2)へいく。

このアルゴリズムにより，適用しようとする問題関数を最大あるいは最小にするようにパラメータが求められるというわけです。問題関数に適/不適があったり，また，遺伝子表現に直すところに職人芸的な技巧が必要とされる場合もありますが，とにかく，今まで使われてきたアルゴリズム(シミュレーテッドアニーリングやランダムリサーチなど)より，このアルゴリズムのほうが解の探索性能がよいといえる問題領域が，確実に存在することが明らかになってきたのです。この実用的な御利益こそが，まず，多くの人に注目されるきっかけを作ったといえるのです。

特別な技巧などを用いないこのアルゴリズムが，なぜうまくいくのでしょうか？その本質は突然変異ではなく，交叉のほうにあるようです。簡単に述べるならば，あるよい遺伝子のよい部分どうしを組み合わせれば，その遺伝子はもっとよくなるだろうと期待できる（「積木仮説」が成り立つともいいます）ような問題において，交叉がうまく働き，単にランダムに探すよりも

発見がうまくいくのです。

ところで，先に示したアルゴリズムの字面上は交叉と突然変異が同列に扱われているようにも見えますが，突然変異の持つ役割は実はきわめて小さいのでは，という考えが主流になりつつあるようです。実際，この春にGoldbergが日本に来たときに2度ほど講演を聞きましたが，たしかに彼の説明でも，突然変異はあくまでもサポート的な働きをしているように扱っていました。

遺伝子アルゴリズムの使い方

GAを実際に体験してみるにはどうすればいいのでしょうか。基本的なアルゴリズム自体をプログラムにするのはそう難しくないでしょうが，いろいろなパラメータを設定可能にして，初めて実用になるという面もありますから，やはりいちばん楽なのは人の作ったプログラムを使ってみることでしょう。

たとえば，最近のインタフェイス（文献3）に積木問題へのGAの適用についての記事が載っており，そのためのプログラムリスト（PC-9801，MS-C対応）が載っています。これならば，自分のパソコンに移すのも，そう難しくないかもしれません。

もっと本格的に取り組むためのシステムとしてGENESISがあり，このカリフォルニア大学版であるGAuscd（標準的なUNIX対応）が国内でも手に入るようになっています。GAuscdには親切なユーザーズマニュアルや例題がついてきます。

添付されている例題のなかに，「$x^2+y^2+z^2$」の最小値を求めるというものがありました。x，y，zはそれぞれ−5.12から5.11まで0.01刻みで値をとるとします。この場合，求めるべき答えはもちろん$x=y=z=0$のときの値0となります。

施行回数，遺伝子総数，突然変異の確率などなど，多数のパラメータを設定して実行させると，第1世代（100回施行後）で関数値のベストが約1.69だったのが，19世代（1042回施工後）で関数値0.0867まで小さくなるという結果が出ました。もちろん，この例は人が見ただけでわかるような簡単な問題でして，複雑な問題でこそGAが本領を発揮するというものでしょう。

人工生命研究のためのGA

ここまでは実用的な側面である最適化アルゴリズムとしてのGAについて説明してきました。これに対し，もともと生物の進化過程のエッセンスを抜き取ったものである遺伝子的アルゴリズムを出発点として，もっと，生命の起源や進化の過程を深く探求していこうとするアプローチも盛んになってきました。これこそがGAの持つ実用的追求とは正反対の側面である「人工生命のためのGA」といえましょう。

そのような側面を追求するひとつの有力なアプローチが，GAとニューラルネットワーク理論の融合であると考えられます。

生物は生まれてから成長するに従って自然界の状況に応じて学習します。また一方，もっと時間軸のメモリを大きくとると，種の進化ということにより，巨視的な意味での学習をしています。前者がニューラルネットワークに対応し，後者がGAに対応すると考えることができます。そして，この両者が互いに補完してきた結果として，現在の生物のレベルに達していると考えられるのです。

このような2つの学習のメカニズムがなぜ必要なのかという問いに対しても，それなりには答えられるかもしれません。もし，一個体における学習しかなく，遺伝がないのならば，生まれたばかりの時点における個体の生命力が弱すぎるということが考えられます。一方，学習がなくて遺伝だけに頼るとすると，環境が変化してくることに対応できなくなりますし，そもそも遺伝情報が莫大になりすぎるかもしれません。

ニューラルネットとGAを融合させるという試みはすでに始まってきています（文献4など）。従来のニューラルネット研究においては，神経細胞間の接続は人間がエイヤっと思いつきで決めて，それでパターンを分類させたりして問題を解くというのが常道でした。しかし，GAを導入すると話が違ってきます。神経細胞間の接続自体もパラメータ化することにより，構造自体も自動的に学習してくれるようになるのです。

しかし正直いって，僕は彼らの研究成果を見て，まだまだスタート地点についたばかりだなという感想を持たざるをえませんでした。ごく簡単な構造を決めるのにもずいぶんと交叉を繰り返す必要があるのです。とてつもない計算量を必要としているのです。どうも，このままではうまくいきそうには思えません。なにかもっと別の，GAと両立するような普遍的な原理の導入が必要なのでしょう。まだまだ困難な問題はありそうですが，狙っているテーマは実に壮大です。風呂敷はとりあえず，精一杯大きく広げましょう，ってところです。

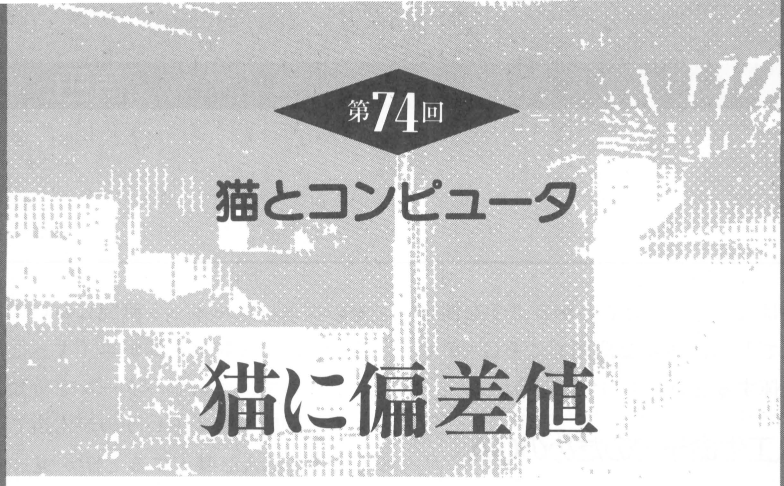
参考文献

1）星野力，遺伝子的アルゴリズム［1］その信仰と現実，bit，1992年9月号，15-25pp.

2）D.E.Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

3）平野広美，積木問題を遺伝子アルゴリズムで解く，インターフェース，1992年2月号，108-137pp.

4）D.Whitley and T.Hanson, Optimizing Neural Networks Using Faster, More Accurate Genetic Search, ICGA'89, 391-396pp.



第74回

猫とコンピュータ

猫に偏差値

Takazawa Kyoko
高沢 恭子

昔から「猫は家につく」などといわれますが，キョウコさんちのご近所には，なんと「住みかえ」をしてしまった猫がいます。今月は，猫と人間，そして人間どうしのコミュニケーションのお話です。

バルセロナと甲子園の余熱にしては，ずいぶんきびしい残暑だった。9月になって観測史に残るという暑さの数日があり，そのあと，にわかに秋が訪れた。

高校野球は，トオルの通う高校が東東京大会でベスト16まで残り，わが家ではニュースだった。音楽部員の彼としては，急造の応援団ながら，第4戦目に二松学舎大付属に敗れるまで青空のもとで選手と一体になって奮闘し，それなりの感動を味わったらしい。私にとっても至近距離での高校野球となり，そのせいか甲子園での明徳義塾－星陵戦はほかの人の3倍くらい大騒ぎしてしまった。

住みかえの時代

とけたアメのようにノビきって，つらい日々をすごしていたホンニャアにも安らぎの季節がきた。ただし，このごろなじみの猫友だちが姿を見せなくなった。

いつも夕方，団地の前でノラ猫に食べ物を与えていた母娘も，管理人に注意を受けたそうで，あらわれなくなった。猫たちが駆除されたとは思えないから，もしかしたらあの母娘は別の場所で食べ物を与えていて，常連のメンバーもそちらに流れていったのかもしれない。

ホンニャアの友人はいまのところ影をひそめてしまったが，引っ越した主人に見切りをつけて，よその家のペットにおさまった猫がいた。

ご近所のカンダさんに宅配便をあずかっていただいたので，恐縮しながら引き取りにうかがったところ，夫人より先にあらわれたのが，なんとも可愛いヒマラヤン種の猫だった。

長いふさふさしたグレイの毛，まん丸の瞳。そばまできて，挨拶をするようにきちんとすわった。

「わぁ，かわいい！」と思わず声をあげたら，つづいてあらわれたカンダ夫人，「ほらネ，みなさんがカワイイっていってくれるでしょ。この猫わかるらしくて，お客さまが見えるとかならず出てくるの。メス猫のせいかしら」

「猫を飼ってらしたんですかぁ」

ご夫妻だけの暮らしだが，夫人は外出がちだし，動物がいるようにはとても思えなかったのだ。

「それが，この並びのヤマモトさんの猫だったのよ」

ヤマモトさんのお宅は日中ほとんど留守で，ひとりぼっちの猫はいつもカンダさんのベランダに遊びにきていたそうだ。食べ物を与えるうちにご夫妻によくなつき，長い時間カンダさんの家ですごすようになった。いごこちがよいらしく，だんだん夜も泊まるようになり，飼い主の家にはたまに帰るだけになった。そのうちヤマモトさんは「お宅にいるほうがシアワセみたい」といいながらどこかに引っ越してしまい，猫も相性のよい家にリハウスしてしまった。

猫というのはみんな警戒心のかたまりで，かんたんに人になじむものではないと思っていたのは，少しちがうようだ。

いつもゆだんなくツマ先で歩き，来客時にはみごとに消えてしまうホンニャアと，長毛の足でゆったり歩き，チャイムの音で来客を出迎える猫では正反対だ。しかもカワイイといわれるのが快感らしい。

キュートな彼女

京大霊長類研究所のチンパンジーのアイちゃんが，1から9までの数を理解しているらしいという記事が新聞にあった。

飼い主が虚構として考えるのでなく，猫自身が「カワイイ」といわれてほんとうに喜ぶとしたら，アイちゃんの「数の理解」と同じほどの驚きだ。数もむずかしいが，美醜への興味はとても高度な関心と思えるからだ。

名前を聞くのを忘れてしまったが，ヒマラヤンの彼女は，はたして「カワイイ」の意味がわかっているだろうか。

そこで推論をひとつ。

動物はごほうびが目的で芸をすることもあるが，何ももらえなくてもけっこうサービスしてくれるものだ。

ホンニャアもわがまま猫だが，無報酬でいろいろなことをやってくれる。カーテンの陰からとびだす「イナイイナイバー」やしっぽをふりまわす「しっぽピンピン」，ほかに未公開の「ヒコウキブーン」というのもある。

角砂糖もビスケットももらえなくても，あんがいみんな人間とふれあうのが楽しみで，自分がワッともてはやされるのが本能的に心地よいのかもしれない。先日，外国のどこかの遊園地から逃げ出したイルカが，ある入り江に迷いこんだときのこと。見物人があつまってきたのを見て，イルカはつい得意の宙がえりをしてしまったそうだ。反射的な行動だったかもしれないが，けっこう本人(?)もかっさいを受ける晴れがましさが好きなのだろう。

おそらく，生まれつき可愛いヒマラヤンの彼女は，自分を見たとたん，とくに女性たちが「きゃあ，カワイイ！」ともてはやすあの瞬間のウケかたが，わけもわからず好きなのだ。彼女の芸は，ただ彼女があらわれることなのだ。

すこし前に，『ネコの偏差値』(注)という本を読んでみた。飼い主が自分の猫を項目

別にチェックしながら，合計点で偏差値を判定するものだった。ユーモアで書かれた本らしく，笑いながら楽しむスコアブックという感じだ。

全体に擬人化されたシニカルな内容で，けっきょく，すなおで警戒心や反抗心の弱い猫ほど偏差値は低く判定される。ちなみにホンニャアは７段階のうちの４番目，ごく平均的な猫という判定だった。

ヒマラヤンの彼女はじつは「カワイイ」の意味もわかる偏差値の高い猫なのかもししれないが，なんのこだわりもなく別の家のペットになってしまうあたりは，前述の本によれば，すなおすぎて偏差値も平凡としか思えない。

猫の性格は飼い主がつくるそうだ。ホンニャアがわがままで生意気なのは私たちのせいだ。ヒマラヤンはもともと温厚な性格だと本にあるけれど，誰とでもなかよくできるのは，飼い主だったヤマモト家の影響もあったと思う。

ただし誰にでもなついてしまうペットというのは，飼い主の庇護欲や責任感を失わせがちだ。猫と飼い主があっさり別れられたのは，おたがいの感化によるのだろう。

プロの雑談に学ぶ

「へえ，読みやすい本だね」

いよいよはじまった公立校の土曜休み。朝食のあとにトオルがパラパラとめくっているのは，最近ソフトバンクから出版された『雑談・パソコン通信論』という本だ。著者は川村清さんと小牧自行さん。どちらも私たちがとてもよく存じあげているかたたちだ。

ともにプログラマであり，テクニカルライターでもあるご両人は，初期のころからパソコン通信の世界に関わりを持ち，その全体や部分を発端からみつめてきた。

パソコン通信は，最近いちだんと多くの人たちに利用されるようになった。システムはパソコンや通信用モデムが支えていても，通信ネットの現場は，おもに文字による情報交換と会話，交流で成り立っている。コンピュータの利用法としては，いちばん日常に近く，人間くさいものだと思う。

それが，一般の人たちにも関心を持たれるのだし，プロフェッショナルの人たちも興味をおぼえるのだろう。

パソコン通信はそんな人間関係をふくめた成り立ちのために，じつにたくさんの要素や表情を持っている。そして，その全容を語りつくすのはとてもむずかしいし，語れる人もすくない。

本誌８月号にも，荻窪圭氏が「パソコン通信に未来はあるか」を書かれていた。通信解剖学のようなメスさばきは，とても読みごたえがあり楽しかった。

パソコンで通信がおこなわれるとき，機械はどんなことをしているのか。あるいはしていないのか。文字コードが言葉に置き換えられて人間に作用するものを，どうあつかうことがよいのか。どんな有利，不利があり，今後どんな期待があるのか。客観的の姿勢でパソコン通信をまんべんなくみわたしたものだった。

『雑談・パソコン通信論』は，対談のかたちで書かれている。こちらは荻窪氏の鋭さとは対照的で，ゆったりと談笑しながら，パソコン通信のさまざまな面について意見を述べあっている。

いくつかのネットの開発や育成を手がけてきたご両人は，メンバーでもありながら，内部からシステム側の当事者としてたくさんのことに出会ってきた。

話は７～８年前からのあれこれなので，とても盛りだくさんだ。これから通信をはじめようとする人にとっても，すでに長く楽しんでいる人にとっても，なかなか興味深い話だと思う。

たとえば，これから手がけようという人には，パソコン通信はどうやってやればよいのか。どんなところが面白い点なのか。ネットにはどんな人がいて，どんな話をしているのか。ホスト局をひらくにはどうしたらよいか。シスオペとしてどんな人が望ましいか。逆に不向きな人とはどんな性格なのか？　そんなことを対談の中でみんな聞かせてくれる。

もう通信を利用している人たちにも，ちょっとした苦労ばなしや，爆笑するようなエピソードが面白い。ほんとはこの実話の数々がいちばん聞きがいがある。

ナツメネットで，システムが完成してもスタッフのほかにだあれもいなかったころの話。ひとりで何役もやって書き込みをしてまわったそうだ。また，わがFBIのシスオペ氏が，一時期多く出没したいわゆる「悪質ゲスト」に，積極的に対面をこころみた話もはじめて知ることだった。

illustration：Kyoko　Takazawa

初心者がうまくネットになじむコツや，ネット内で起こりがちなトラブル，人間関係。ハンドルネームにまつわる話。ハプニング，ハッピネス，などなど。

対談の構成なので，どの部分からでも話に入りこめて，理解しやすい。それが，トオルも読みやすいといった理由なのだろう。

そしてたくさんの具体的な話を披露しながら，これからのパソコン通信をどう成長させていったらよいかという課題にせまっていく。

パソコン通信とひとくちにいっても，利用する人の目的によって，理想のかたちはいろいろあるだろう。私なりにいくつかの目的で利用はしていても，もっと有効なパソコン通信がどんなものかと考えると，積極的な意見はとぼしい。

この『……通信論』の最終章は「パソコン通信を“過去のもの”にしないために」というものだ。

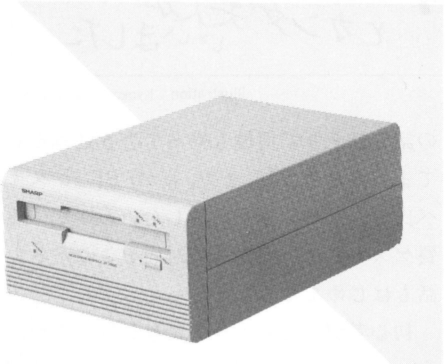
プログラマのご両人が，あえて専門的な分野にあまりふれないで，パソコン通信の人間模様をメインに語っているところに，私はパソコン通信への希望を感じているのだけれど。

（注）『ネコの偏差値』
ピーター・マンデル/安生和之著
相原真理子訳　講談社刊

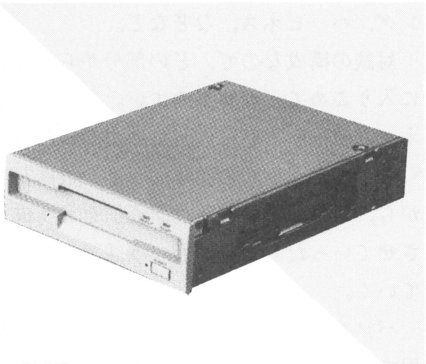
PENGUIN INFORMATION CORNER
ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

ハーフハイト光磁気ディスクドライブ
JY-7500/750
シャープ



JY-7500



JY-750

シャープは，光磁気ディスクドライブ「JY-7500/750」を発売する。

本機は，従来機に比べ回路の高集積化を図り重量は3分の1，発熱量は2分の1以下に抑え，より信頼性を高めた。

インタフェイスにはSCSI，SCSI2が使用でき，平均データ転送速度は，セクタ長512バイトで793Kバイト/s，1024バイトで870Kバイト/s，平均シークタイムは40msと高速化がなされている。

このドライブでは，ISO規格準拠の5.25インチ書き換え型光ディスクユニットと，オプションでヒューレット・パッカード社仕様の追記型ディスクも使用できる。

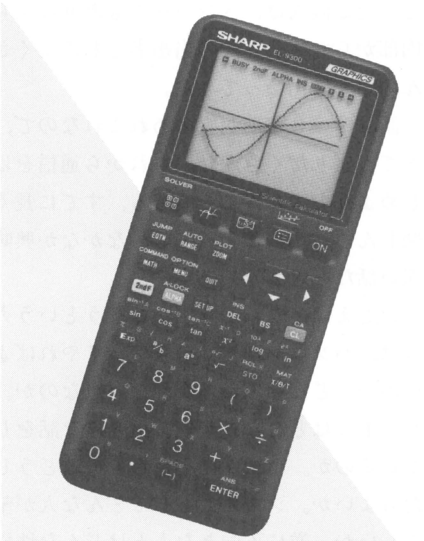
サンプル価格は「JY-7500」が350,000円，ディスクドライブユニットのみの「JY-750」が300,000円（ともに税別)。

〈問い合わせ先〉

シャープ㈱　☎03(3260)1161，06(621)1221

音声電卓・グラフィック関数電卓
CS-2600/EL-9300
シャープ

CS-2600



EL-9300

シャープは，音声電卓「CS-2600」と関数電卓「EL-9300」を発売した。

「CS-2600」は，社会福祉法人の日本盲人連合と日本点字図書館を通じて販売されるもので，視覚障害者が利用できるように音声機能を備えた電卓である。

音声機能としては，数字，四則演算などのキー操作に応じて，キー名称の読み上げや表示している数値の読み上げを行うものがある。また，位読み機能も備え，計算経過や結果を音声で確認できるようになっている。

「EL-9300」は，96×64ドットマトリクスハイコントラスト液晶画面を使い，複雑な式も記述どおりに表示できる関数電卓である。

グラフ描画機能も備え，直交座標，局座標媒介変数グラフの座標系を使用でき，最大で4つのグラフを同時に描画することもできる。さらにカード型データ入力画面を持つ統計機能もサポート。それぞれの機能を簡単に選択できるように，メニューシステムを採用し，大きな液晶画面を生かした高機能な関数電卓となっている。

ユーザーエリアとして約23KバイトのRAMを搭載し，分数計算，数値積分，行列演算など286関数を使用できる。

価格は，「CS-2600」が52,000円(税別)，「EL-9300」が19,800円（税別）である。

〈問い合わせ先〉

シャープ㈱　☎03(3260)1161，06(621)1221

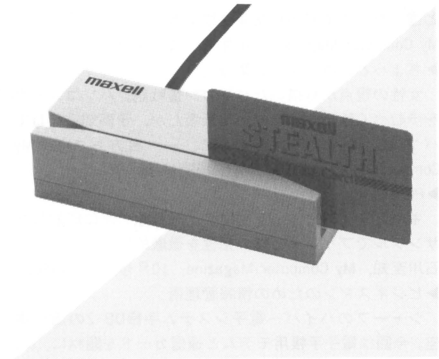
特殊蛍光体で印刷された
ステルス型バーコード
日立マクセル

日立マクセルは，ステルス（不可視）型バーコード用材料の開発，読み取り装置の開発に成功した。

従来のバーコードでは，商品パッケージにはっきり印刷されるため，その商品の美観を損ねたり，読み取りの性格上汚れに弱いところがあった。このバーコードは無色透明に近く，さらに，赤外光を使って読み込むため，汚れに強い耐性を示している。

さらにバーコードの上に通常の印刷を行っても読み取りができ，デザインを重視した商品でのバーコード管理，POS用，FA用としての利用が期待される。

また，ステルス型バーコードリーダは，分解能0.4mmの挿入式とスロット式の2



種類を開発済み。現在，分解能0.2mmのタッチ式，ペン型も開発中である。

カードリーダの本体価格は未定。

〈問い合わせ先〉

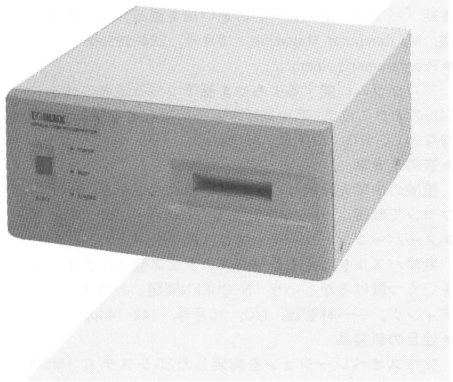
日立マクセル㈱　☎03(3241)9733

光カードリーダ/ライタ

LC-304

日本コンラックス

LC-304



日本コンラックスでは，光カードリーダライタ「LC-304」を開発した。

「LC-304」は，DELA規格の追記型光カードを読み書きするためのリーダライタである。

書き込み速度，読み込み速度はともに160Kバイト/sを実現，アクセスタイムも隣接トラックで5.0msと高速に読み書きができる。インタフェイスにはSCSI，RS-232Cを使用し，現在使われている多くのパソコンへ接続が可能となっている。

追記型光カードはユーザー容量として2.8Mバイト，全容量として4.1Mバイトの記憶容量を持ち，1枚当たりの価格は1,000円以下（量産時）で製造ができる。

カードの大きさは，従来の磁気カードと同じ大きさで携帯性にもすぐれている。また，大容量のメモリカードとして，実用化に向けさまざまな分野で開発が進んでいるようだ。

本体価格は500,000円（税別）となっている。

〈問い合わせ先〉

㈱日本コンラックス　☎03(3502)1811

INFORMATION

電脳未来展

TRONSHOW'92

トロン・ショウ実行委員会

トロン・ショウ実行委員会は，12月1日(火)～2日(水)の2日間(10：00～17：30)，東京青山のTEPIAで「TRONSHOW'92」を開催する。

本イベントは，1990年から毎年開催され，現在のTRONプロジェクト活動状況，製品化動向をトータルに紹介することを目的として行われるものである。

展示内容はTRONプロジェクトの紹介，TRONプロジェクト関連製品，TRON応用プロジェクトなどとなっている。

また，電脳社会における未来社会を考える電脳未来都市展，コンピュータによる身体障害者支援を考えるイネーブルウェア展も併催する予定である。

入場料は1,000円（予定）。

〈問い合わせ先〉

トロン・ショウ実行委員会事務局

☎03(3783)3101

論文コンテスト

多様化・飛躍化するPC環境を考える

化成バーベイタム

化成バーベイタムでは，パソコン，ワープロユーザーを対象として，情報，パソコンに関しての論文コンテストを実施する。

募集テーマは，

1)　これからのパソコン環境を考える
2)　私のパソコン情報管理技術
3)　私にとってのパソコン
4)　いままでにないこんな周辺機器が欲しい

以上，4テーマでパソコンの環境に新しい可能性，創造性を感じさせるもの。

応募方法は400字詰め原稿用紙5枚以内にまとめ，最終ページに応募者の氏名，年齢，住所，職業を明記して下記の住所まで郵送で送付する。

また，選択テーマを変え，ひとりで複数の応募も可。応募締切は1993年2月1日（当日消印有効）となっている。

〈応募先〉

〒105　東京都港区芝2-1-28　成旺ビル

化成バーベイタム(株)論文コンテスト係

〈問い合わせ先〉

化成バーベイタム(株)　☎03(3563)2359

エンタテイメントスクエア

チルコポルト

コナミ/コナミエンタテイメント

チルコポルト入口

コナミは，神戸ハーバーランドモザイク内に大型アミューズメント施設の第1号店として，エンタテイメントスクエア「チルコポルト」をオープンした。

施設は神戸ハーバーランドモザイク内の2～4階にあり，2階は体感シミュレーション，ビデオコーナー，ファーストフードの飲食施設を設置，3階はメダルカジノ，カウンターバー，シネマとアミューズメントのラウンジスペースの間にカフェテリアを設置している。

4階は3階を見下ろすブリッジ通路を渡り，本格的な対人ディーラーコーナーとビリヤードを楽しめるようになっている。また，4階にはディスクジョッキースタジオを設け，DJサービスも行う予定。

営業時間は，11：00～23：00（年中無休），ただし飲食関係は11：00～22：00となっている。

〈問い合わせ先〉

コナミ(株)　☎03(3432)5668

FILES Oh!X

このインデックスは，タイトル，注記——著者名，誌名，月号，ページで構成されています。寒くなってきました。北国の人はもう冬の準備をしているのかな。食べ物もおいしいし，紅葉も美しい行楽シーズンです。みんな，健康第一でがんばってくださいね。

参考文献
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
テクノポリス　徳間書店
PIXEL　図形処理情報センター
POPCOM　小学館
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶THE NEWS FILE
新・技術家庭教科書に導入された「パソコン授業」やCGコンテストの話題など，パソコン周辺のゴシップやインタビュー。——編集部，LOGIN，18号，36-43pp.

▶特集　ゲーム考古学
パソコンの歴史はゲームと共にあり。打ち込みマニアから始まったパソコンゲームの歴史と文化をおもしろおかしく分析解説。——編集部，LOGIN，18号，211-225pp.

▶MUSIC LABO
ゲームミュージックファンから，パソコンを楽器として操りたいMIDI初心者までへのパソコンミュージックのページ。連載第1回の今回はMIDI最新楽器やソフトを紹介。——編集部，LOGIN，18号，236-243pp.

▶Hardware Laboratory
マルチメディアマシンCD-Iが各社から発売され始めた。今回はフィリップスのCD-Iプレイヤー「CDI-205」を紹介。——編集部，POPCOM，10月号，114-116pp.

▶アルゴリズムを見切ったぞ!?
今回はレースゲームのコースを3D表示するアルゴリズムを考える。道を変形させる処理のアルゴリズムを紹介。——おにおん，テクノポリス，10月号，136-140pp.

▶パソコン通信大特集
パソコン通信に興味はあっても手を出せないでいる人へ。読んでためになるハードやソフトの解説記事。——編集部，マイコンBASIC Magazine，10月号，59-80pp.

▶BASICプログラミング講座　第6回
物理シミュレーションの基本中の基本，放物線運動をゲームにしてみよう。コンピュータでの計算の基礎を，ゲームを作ることで学ぶ。——東　幸太，マイコンBASIC Magazine，10月号，104-108pp.

▶平成環境委員会
パソコンの配置やおそうじグッズなど，パソコンの物理的な環境をもっと快適にするための特集。セッティングの知恵を伝授し，編集スタッフの机の写真も公開。——編集部，ASCII，10月号，205-227pp.

▶コンピュータ帝国の興亡
米国パソコン業界の誕生と発展の歴史をつづった単行本『ACCIDENTAL EMPIRES』より，ビル・ゲイツのエピソードを抜粋して掲載。——Robert X.Cringely/アスキー書籍編集部，ASCII，10月号，229-236pp.

▶失敗しないプリンタ選び'92
各種プリンタの特徴や用途別プリンタ環境，最新機種の紹介などを通じ，目的に合わせて何を選ぶべきかを考察する。——編集部，ASCII，10月号，245-260pp.

▶CD-ROMって，こんなに便利
78,000円という戦略価格で発売のNECのCD-ROMドライブセット「PC-CD50S」や，電子ブックプレーヤーの新製品情報，パームトップから電子ブックへのアクセス方法など。——志村拓，ASCII，10月号，285-292pp.

▶SIGGRAPH'92
SIGGRAPHはACM（米国計算機学会）のコンピュータグラフィックスに関する分科会である。そこで発表されたCG作品を中心に，今年の内容報告と今後の動向についてレポートする。——編集部，ASCII，10月号，301-308pp.

▶ことば遊び・コンピュータ
言葉遊びがテーマの連載。今月は駄洒落プログラムを作る。母音や子音の置き換え，音節の入れ替えなどで，似た表現を探す。「コッペリアのビジャヤナガル（河童の川流れ）」はコンピュータならではのセンス？——ホーテンス・S・エンドウ，ASCII，10月号，333-336pp.

▶ラッキー！ハッピー！オッケー！
コンピュータを使うときに発生する法律上の問題を考えるページ。今月は，CD-ROMソフトを引用して会社の書類を作れるか，など。——編集部，ASCII，10月号，384p.

▶マイコンからMy Computerへ
マイコンピュータ・マガジン創刊15周年記念企画。マイクロソフト代表取締役社長成毛真氏，セイコーエプソン国内コンピュータ事業部の内田健治氏らにインタビュー。——編集部，My Computer Magazine，10月号，88-103pp.

▶今どき一万円で，なにが買えるか？をフィールド実験
秋葉原のショップを訪ね，1万円でお買い物。ちょっと違った角度で贈る秋葉原タウンガイド。——石川至知，My Computer Magazine，10月号，162-167pp.

▶ちょっとだけパソコン気分
女性の視点から描いたパソコン奮戦記。パソコン人生を歩むべく秋葉原に出向いた筆者だが，予備知識なしでパソコン購入はできるのか？——名方ちさ子，My Computer Magazine，10月号，168-171pp.

▶PC実験室
キヤノン「BJ-10V」を題材に，30種類の紙に印刷したサンプルでプリンタと紙の相性を徹底的に調べる。——石川至知，My Computer Magazine，10月号，172-178pp.

▶ビジネスマンのための情報管理術
シャープのハイパー電子システム手帳DB-Zの活用講座。今回は電子手帳用モデムと通信カードを題材に，DB-Zの携帯性を活かしたアクセス方法を紹介する。——塚田洋一，My Computer Magazine，10月号，238-241pp.

▶郵政業務に生かされるコンピュータ技術
大量の郵便物を効率的に仕分けするための，スキャナを使ったシステムの素顔と仕組みに迫る。——野沢潤一郎，My Computer Magazine，10月号，254-257pp.

▶ナムコ・ワンダーエッグ
東京・二子玉川園にあるアミューズメントパーク「ワンダーエッグ」のアトラクションの内容などをレポート。——野沢潤一郎，My Computer Magazine，10月号，270-273pp.

▶MYCOM WATCHING
「海から空をにらむ護衛艦」と題し，海上自衛隊のミサイル搭載護衛艦DDGを取材する。護衛艦の装備の紹介と空対艦ミサイルへの対処のテクノロジーを解説。——菊地秀一，My Computer Magazine，10月号，274-278pp.

▶マイコン考古学
元ドクターパソコン宮永好道氏が贈るパーソナルコンピュータ回顧録。シャープのMZ発表当時を振り返り，TV番組「パソコンサンデー」の思い出を語る。——宮永好道，My Computer Magazine，10月号，294-298pp.

▶Programmer's Room
プログラムに関するよもやま話でつづる読者のページ。DOSのテクニック，アルゴリズムの解説などを掲載。——ななふし，I/O，10月号，85-88pp.

▶夜の走光車
暗闇の中で光を頼りに進む装甲車ロボットをノートパソコンで制御。——ASUKE，I/O，10月号，128-136pp.

▶スーパーコンピューティング入門
数学パズルを考えるシリーズ。チェス盤上にクイーンをいくつ置けるかという「N-QUEEN問題」の解法とコーディング。——林智雄，I/O，10月号，142-144pp.

▶注目の新製品
マウスオペレーションを実現したCGシステム「MIRAGE」を紹介。——松永　忠，PIXEL，10月号，160-162pp.

MZシリーズ

MZ-1500（BASIC MZ-5Z001）

▶狙撃せよ！
人形（ひとがた）の的のうち，拳銃を持つもののみを撃つ。判断力が身につくシューティング・アクションゲーム。——舟生日出男，マイコンBASIC Magazine，10月号，128-129pp.

MZ-2500（BASIC-M25）

▶FINAL BOOST
さまざまなコースを制限時間内で走り切れ。ドリフト走行が楽しいレースゲーム。——GODDESS'EYE，マイコンBASIC Magazine，10月号，130-132pp.

X1/turbo/Z

X1シリーズ

▶妙人変態やぎおくん
嫌われ者のやぎおくんを動かし，追いつめた女の子を泣かすと面クリア，のアクション・パズルゲーム。——平光利浩，マイコンBASIC Magazine，10月号，155-156pp.

X1turboシリーズ

▶卵を守れ！げーむ
ケツ頭星人から「星の卵」を守れ。アクション・シューティング。——へる，マイコンBASIC Magazine，10月号，157-158pp.

X68000

▶NEW SOFT
PCエンジンから移植された「ネクタリス」などの新着ゲームソフトを紹介。――編集部，LOGIN，18号，30p.

▶最新ゲーム徹底解剖!!
遂に登場した「ポピュラスⅡ」。まずは基本事項の解説。個性豊かな6体のロボットで敵ロボット軍を破壊せよ！「シュートレンジ」。「ファイナルファイト」のステージ別完璧攻略法。――編集部，LOGIN，18号，126-173pp.

▶X68000新聞
硬派なSF戦闘シミュレーション「ネクタリス」，格闘ゲーム「デスブレイド」，開発中の「リゾート奪回作戦（仮）」，アクションRPG「サークII」。――編集部，LOGIN，18号，248-251pp.

▶Software Hot Press
「ネクタリス」，ゲームエディタ「シューティング68K」による作品集「シューティング68K GAMES」，人間と怪物たちのデスマッチ「デスブレイド」，「エトワールプリンセス」。――編集部，POPCOM，10月号，16-25pp.

▶SOFT EXPRESS
新作・開発中ゲームソフト紹介。「バーンウェルト」など。――編集部，コンプティーク，10月号，66p.

▶GAMING WORLD
「ネクタリス」，「エトワールプリンセス」，「サークⅡ」などを紹介。――編集部，テクノポリス，10月号，34-42pp.

▶燃える攻略野郎
X68000版も開発中の，築城シミュレーション「キャッスルズ」をPC-9801版で攻略。――編集部，テクノポリス，10月号，66-68pp.

▶PONOMU
何も考えずに歩いている「ぽのむ」を扉へ導いて！　レミングス風アクションパズル。――青木賢太郎，マイコンBASIC Magazine，10月号，159-160pp.

▶MOUNTAIN PEAK
ベストタイムを競うカーレースゲーム。――高橋秀之，マイコンBASIC Magazine，10月号，161-163pp.

▶ターボアウトラン　～Keep Your Heart～
セガのゲームミュージックプログラム。要NAGDRV。――完コピD.O，マイコンBASIC Magazine，10月号，173-175pp.

▶今月の注目パソコン・ゲーム
新着ゲームの紹介。「ライジングサン」，「ネクタリス」。――いたばし・猪野清秀，マイコンBASIC Magazine，10月号，227-228pp.

▶AV STRASSE
エル・クラフトの通信機能付きスクリーンエディタ「Xe Ver.Ⅱ」，「DōGA CGAシステムVer.2.5」，M.N.M Softwareのスプライトエディタ「ぴくせる君 Ver.1.20」などを紹介。――編集部，ASCII，10月号，317-320pp.

▶HI-TECH HIKERS
ETとX68000のデータ交換を可能にするフリーソフトウェア「ETran.X」を紹介。システム手帳の用途を広げる1本だ。――編集部，ASCII，10月号，348p.

▶TBN SPECIAL
「バトルテックセンター横浜」が横浜TRELL-ONEにオープン。ヴァーチャルワールドの中でロボットどうしの戦闘を行うバトルテックの世界をX68000用ソフトウェアもからめて紹介。――編集部，ASCII，10月号，358-359pp.

▶FREE SOFTWARE INDEX
ここ数カ月の間に主要BBSにアップロードされたソフトウェアをピックアップして告知する。X68000用ソフトは，テキストエディタED.R，RCPファイルプレーヤーSXRCP.Xなど。――編集部，ASCII，10月号，402-403pp.

▶長期ロードテスト
X68000EXPERTⅡの報告第11回。TeXシステムのインストールから印字に至るまでの試行錯誤の模様をレポートする。――編集部，ASCII，10月号，415-417pp.

▶なんでもQ＆A
X68000使用中に発生する疑問にメーカー自身が回答する。新製品CHART PRO-68Kの特長や，Multiwordで書式設定を変える方法，SX-WINDOWの背景にカラーを使用できるか？　など。――シャープ株式会社AVCシステム事業推進室，My Computer Magazine，10月号，336-337pp.

▶HOBBY EXPRESS
X68000用アクションゲーム「ファイナルファイト」のストーリーと魅力，出来具合を評価する。――あゆさわかつみ，My Computer Magazine，10月号，362-363pp.

▶高速日本地図エディタ68
PC-9801用に発表されていた高速日本地図エディタの移植版。地図の中から都道府県や都市の検索ができる。――新妻幹也，I/O，10月号，106-107pp.

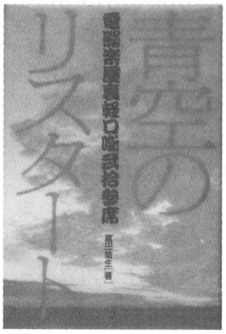
▶bmp68
MS-WINDOWSの標準画像フォーマットのファイルをX68000で読み込むためのユーティリティ。DIB形式からPIX形式への変換ができるので，SX-WINDOWへのグラフィックコンバートも可能。――大澤文孝，I/O，10月号，119-121pp.

ポケコン

PC-E500

▶ダッシュマン
ダッシュして，崖の直前ギリギリのところで止まる。崖寸前まで行けばOK。落ちればハイそれまでよ。いろんなタイプの地面が用意されている神経すりへらしゲーム。――近藤紀之，マイコンBASIC Magazine，10月号，165p.

新刊書案内



青空のリスタート
富田倫生著
ソフトバンク刊
☎03(5488)1360
四六判　237ページ
1,600円(税込)

こういう文章を書ける人がまだパソコン業界に残っているのだ，とそう思う人は幸いで，なんだ，このふざけたやつは，おれはパソコンの勉強をしたかったんだ，と思う人は不幸である。松田優作の話からはじまってTOWNSへ行ってしまう流れ，アントニオ猪木の叫びがいつのまにかサンマイクロシステムズの話になってしまう強引さ，自虐的でありながらカッコつけた文体。テーマはそれこそTOWNS，Windows，DOS/V，UNIX，Macintosh，TRON，AMDの386互換チップ，ダウンサイジングなどなど広く広く取り扱い（そういえばPC-9801関係の話題はなかったけれど），それをありそうにないことでも独自の視点で断じてしまう潔さが本書のユニークさを際立たせている。

本書はソフトバンクの今はなき『パソコン・マガジン』に連載されたコラムに加筆訂正（というよりも，連載されたコラムの元ファイルを引っ張り出して修正したものらしい），さらに，各項目にうだうだと言い訳であったり後日談であったりを追加してできあがっている，いわばエッセイ集である。著者の富田倫生氏は台所でPowerBookを使って原稿を書くMacintoshユーザーであり，その奥にはパソコンが個人の力を開放する，というパソコン黎明期に西海岸で生まれた思想を抱えている。最近では珍しくなってしまった，個人のためのコンピュータを追求する人であり，今のビジネス全盛パソコン界に真っ向から挑む人だ。テクニカルな内容はどうでもよく，どう使うべきか，どう使われれば面白くなるかが問題。

そういうわけで，エッセイ集でありながら，すべてのコラムが筆者の視点と思想と立脚点から絶対にずれない，一種のパソコン界の解説になっている。パソコン・マガジン連載時からの読者であった身からすれば，これが単行本化されたのは無上の喜びであり，意見が合わないところがあっても許してしまうのである。　（K）

別冊宝島EX
現代数学で遊ぶ本
JICC出版局刊
☎03(3234)4621
A5判　202ページ
1,300円(税込)

「カオス，フラクタクルから火の玉量子力学まで異種交流の現代数学ガイド」という副題だが，本書は，必ずしも「数学好き」の人が対象ではない。むしろ，学校教育のなかで選別の手段として使われてしまっている数学の，ほんとうの楽しさをすべての人に理解してもらおうという本である。「数」とは何か，から始まり，数学者たちの夢の軌跡まで，数学という文化が述べられている。

「情報産業時代の今日，数学は現代の文化の中に無数に埋め込まれているといってもよい。（中略）気おくれせず首をつっこんでいけば，現代社会が見えてくるのである」（まえがきより）

PC-PAGE 28
乱調電脳用語事典
「乱調電脳用語事典」
編集委員会編
翔泳社刊
☎03(5467)0361
A5判　244ページ
1,600円(税込)

副題は「パソコンの明日を考えるためのキーワード52」。パソコンを理解するうえで重要な52項目について，ソフトウェア，情報，ハードウェア，そして文化・歴史の4分野に分けて解説している。

しかし，本書はたんなる用語解説書ではない。それぞれの項目について，その定義づけが述べられたあと，その定義の正当性をめぐって，4人の編者の討論が行われ，最後に補足的な用語解説が入る，という構成である。そして，ユーザーの混乱や疑問がそのままさらけだされたこの討論のなかから，いわゆる用語事典からは得られない，パソコン世界の現在状況が浮かび上がってくる。

QUESTION and ANSWER

Q **1991年7月号のOh!Xで「X-BASICポケットリファレンス」が特別付録にあり，おかげさまでいまでも重宝して使っています。**

このリファレンスの56～57ページに未公開命令が掲載されていたので，それを使ったプログラムをX-BASICで作りました。そして「C Compiler PRO-68K ver.1.0」を使用して，そのプログラムをコンパイルしたときは正常に動作したのですが，同じプログラムを「C Compiler PRO-68K ver.2.1」を使用してコンパイルすると，コンパイル時にエラーは出ないのですが，コンパイルして生成された実行ファイル（～.Xというファイル）を起動すると，画面の中央に“エラーが発生しました”と表示され，正常に動作しません。

このとき使用した未公開命令は【A_END】【A_STAT】【A_STOP】の3つです。

この症状を回避する方法（C Compiler PRO-68K ver.2.1でコンパイルしても正常に動作させる方法）を，ぜひとも教えていただけないでしょうか。

東京都　山口　隆久

A 最初に確認しておきますが，未公開命令とはメーカーで動作の保証をしていない命令です。X68000CompactXVIに付属のX-BASICのマニュアルにも上の3命令は公開されていません。したがって現時点でも，メーカーによって動作の保証はされていません。将来はメーカー保証の命令となる可能性もありますが，逆に命令が削除されてしまうかもしれません。未公開命令を使用することはそれだけのリスクを背負っていることを忘れないでください。

さて，C Compiler ver.2.1にはFM/MIDI/AD PCM音源を制御するドライバとして，従来のOPMDRV2.Xに代わってOPMDRV3.Xが収められています。OPMDRV3.XでAD PCMの制御をするようになったため，従来AD PCMを制御したPCMDRV.SYSは組み込みが不要になりました。これですとPCMDRV.SYSでサポートしていた機能がコンパイルして使えるかどうか心配です。しかし，ver.2.1に付属の拡張マニュアルには「AD PCM用の外部関数(A_PLAY，A_REC)も従来通りお使いいただけます」とありますので，公開命令についてはPCMDRV.SYSの持っている機能も安心して使えるようです。

さて，問題は質問にある3つの未公開命令のコンパイルですが，コンパイルが正常に終了するということはX-BASICの関数を集めたライブラリ(BASLIB.L)には3つの命令ともオブジェクトが含まれているはずです。

ver.2.1に付属のBASLIB.Lの内容をライブラリアン(LIB.X)で調べてみると，3命令ともオブジェクトのあることが確認できました。さらに3つのファイルを取り出してみると，それまでのライブラリがIOCSコール＄67(ADPCMMOD)を使ってAD PCMを制御していたところを，OPMDRV3.Xで拡張されたOPMDRVコールを使うように変更されていました。ライブラリは変更されているものの，上記3つの未公開命令はC Compiler ver.2.1でもサポートされていると考えていいでしょう。

編集室のマシンでは，ver.2.1で未公開命令がコンパイルできましたし，実行もできました。OPMDRV3.Xを組み込んでいれば，A_END，A_CONT，A_STOPとも動作します。OPMDRV.X，OPMDRV2.Xを組み込んでいるとエラーが発生することはありませんが，正常に動作しません。

こうなってくると，山口さんのシステム環境に問題があるようです。チェックしてもらいたのは，

1)　ver.2.1のライブラリを使ってコンパイルしているのなら，必ずOPMDRV3.Xを組み込んで実行すること。

2)　もしもCC.X本体だけがver.2.1でライブラリは旧版のものを使っているなら，OPMDRV.X，OPMDRV2.Xを組み込んで実行すること。

質問にある「エラーが発生しました」というメッセージが気になります。そのメッセージが，

　エラー($01F0)が発生しました

と表示されるのであれば，OPMDRV3.Xを組み込んでいないはずです。(1)，(2)を参考にして正しくOPMDRV3.Xを組み込んで実行してください。

Q **プログラムでRAMディスクのドライブ番号を知りたいのですが，指定したドライブのメディアを調べるにはどうしたらいいのでしょうか。**

東京都　堀江　光太郎

A 2HDドライブ，HD，MOといった媒体は，それぞれに固有の媒体番号がつけられています。この番号をメディアバイトと呼びます。メディアバイトを調べるにはDOSコール$FF32(getdpb)を使います。このDOSコールはドライブパラメータブロック（94バイト）の内容を得ることができますが，そのうちメディアバイトは先頭から＋22バイト目にあります。アセンブラで指定ドライブのメディアバイトを調べる例を示すと，

```
pea.l    dpbptr
move.w   #1,-(sp)    *DRIVE A
dc.w     $ff32       *GETDPB実行
```

表1

媒体	メディアバイト
2HDドライブ	$FE
2DD	$FB
RAMDISK	$F9
SASI HD	$F8
SCSI HD	$F7
MO	$F6
CD-ROM	GETDPBのエラーコードが－14

リスト1

```
================== media.c ==================
#include <stdio.h>
#include <doslib.h>
struct DPBPTR BUFFER;

void main()
{
    int i;
    int result;
    char drv[] = 'A';

    for (i=1; i<=26; i++) {
        result = GETDPB(i, &BUFFER);
        if (result == -15) {
            printf("¥nすべてのDRIVEを調べ終えました¥n");
            exit(0);
        }
        drv[0] = 'A'-1+i;
        if (result ==.-14) {
            printf("DRIVE %s はCD-ROMです.¥n", drv);
        } else {
        printf("DRIVE %s のメディアバイトは $%X です.¥n", drv, BUFFER.id);
        }
    }
}
```

```
    addq.l    #6,sp
    tst.l     d0
    bmi       エラー処理
    move.b    22(a1),d0
```

のようにすると，d0レジスタにメディアバイトが入ります。メディアバイトに対応する媒体装置の種類がわからなければどうしようもないので，編集室のマシンでMOなどのメディアバイトを調べたものを表2にまとめておきました。SRAM_DISKはRAM_DISKと同じメディアバイトでした。

表2からRAMディスクのメディアバイトは$F9なので，上のプログラムの最後のmove.b 22(a1)，d0を，

```
    cmpi.b    #$f9,22(a1)
    bne       RAMディスクでない
                :
                :
    RAMディスクのときの処理
                :
                :
```

とすることができます。

リスト1はメディアバイトを調べるために作成したプログラムです。初めてCで書いたプログラムなので，変なところがあるかもしれませんが笑ってやってください。このプログラムを，

A>CC /Y MEDIA.C

としてコンパイルするとMEDIA.Xが作成されます。実行すると接続しているすべてのドライブのメディアバイトを表示します。

なお計測技研のCD-ROMはGETDPBでドライブパラメータを返してきません。通常，ドライブが接続されていない場合のエラーコードは－15ですが，CD-ROMが接続されている場合はエラーコード－14を返しますので，GETDPBのエラーコードでCD-ROMを認識してください。

（影山裕昭）

Q　新しいFIXERはSX-WINDOWに対応したと書いてありましたが，普通にASKと入れ替えただけだと，カーソルキーとファンクションキーを押したときに，KEY.SYSの内容が出力されてしまいます。どうしたらいいのでしょうか？　　東京都　長谷川　正典

A　結論から先にいってしまうと，FIXERがSX-WINDOWにしっかり対応していないことから起こる弊害です。ディスアセンブルなどをして解析したわけではないので，理由は予想の範囲を出ないのですが，おそらくSX-WINDOW（キーボードマンあるいはキーマン？）のKEY入力関係のベクタのフックに失敗しているのではないかと私は考えています。

完全に対応するには，FXIERをSXWINDOWにしっかり対応してくれるようにメーカーにバージョンアップを望むしかありません。しかし，それではいつになるかわからないので，KEY.SYSの内容をうまく設定し，実用に差し支えない程度の応急処置をしてみました。

困ったことにKEY.SYSの内容って，デフォルトでF6にESC･Eが入ってるんですよね。私はそのために，何度エディタ.Xで全セーブしてしまったことか……。

ダンプリストのKEY.SYSは，エディタ.Xのファンクションキーの内容と同じ動作をするコントロールコマンドやエスケープコマンドを選んで登録しています。6月号付録のMAC.Xから打ち込んでLHA.Xで展開してください（セーブは114バイト）。普通に自分でマニュアルを調べてKEY.Xで登録しても別にかまわないのですが，KEY.Xでは,^Cを入力できないので(エディタ.Xではロールアップ）デバッカなどでいじることができない人は単純に入力したほうが楽でしょう。

ただ，これでまずいのは，エディタ.XはSHIFT＋F5，F5～F10，UNDOなどに配分された機能に相当するコントロール，またはエスケープコマンドが存在しないことです。単純になにもKEY.SYSに入力しないと，そのキーは使えなくなってしまうのです。しかも，F6～F10はカット＆ペースト処理でなくては困る機能です。

そこで，なにか入ってなくてはならないのならば，なにもしないコマンドESC･ESC（実は思いつくのにずいぶん時間がかかった）を入れておけばよいのでした。

最初にいっておくのを忘れましたが，SX-WINDOWでFIXERを使うときには，SXWIN.X起動時に-Kをつけておくのを忘れずに(起動後CTRL＋OPT1＋F10でも同じことができる)。このオプションはIOCSキー入力を有効にするものです。

実はこれだけで，ある程度，本当に少し使っただけではわからない程度，FIXERを使えるのですが，これでもまだ何分の1かの確率でKEY.SYSの内容が出力されてしまうのでした。

なお，あまり関係ないのですが，私のようにいきなりSX-WINDOWからシステムを起動する人はこれでよいかもしれませんが，コマンドシェルからシステムを起動している人は，ヒストリなどが使えなくて困ると思います。

そのときはバッチファイルなどでKEY.Xを起動し，自動的にKEY.SYSをロードするようなリダイレクトファイルを与えればスムースに任意にキー配置を変えることができます。

しかしながら，応急処置はしたものの，FIXERにはメーカーからの（正確な）バグレポートがほしいところですね。マルチワードでも，WP.Xでもまともに使えないことですし。　（瀧　康史）

リスト2

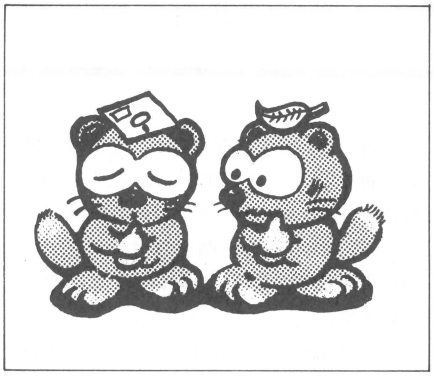
```
0000  1C 0F 2D 6C 68 31 2D 53  : DD
0008  00 00 00 C8 02 00 00 65  : 2F
0010  2B 30 19 20 00 06 4B 45  : 2A
0018  59 2E 53 58 38 9A D3 F3  : CA
0020  B1 94 C0 0C 5F 35 40 7F  : 64
0028  AD 80 1F ED 5D 0F DD D3  : 55
0030  87 EE 43 F6 C3 F2 81 FE  : E2
0038  95 E0 FF 6E C0 FF 62 C0  : C3
0040  FF 7E E1 BF 18 1F E6 D6  : 10
0048  1F E8 56 1F F2 CC 3F E3  : 5C
0050  A0 7F 9F 7B 0F E3 E3 80  : 8E
0058  B9 EB 64 2F 27 4A 0B C8  : 7B
0060  C9 02 E7 E8 01 72 11 81  : 9F
0068  74 93 8E FC B4 E4 6E 76  : 0D
0070  35 00 00 00 00 00 00 00  : 35
0078  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  03 B4 69 75 D6 74 DD F8  2BDC
```

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108　東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部

STUDIO X ON AIR



FROM READERS TO THE EDITOR

行楽シーズン到来！　涼しくなったし，食べ物もおいしいし，身体を鍛えて冬に備えましょう。運動神経に自信がない人はカメラや絵筆を持って出かけるのもいいですね。でも，はりきりすぎて風邪などひかないように，ご注意。

◆えっ，E.O.さんが音楽一筋に生きるために引退するんですか(……ちょっぴり涙)。彼女の文章からにじみ出てくる人柄が大好きだったんです(……やっぱり涙)。とても残念ですが，これも人生，E.O.さん，がんばってくださいね。えっ結婚なさるんですか(大粒の涙)。

石田　智義(21)京都府

◆そうですか，E.O.さんご結婚されるのですか。私はとてもあいまいな人間なので，E.O.さんの文章の強さには気押されながら読んでいたものでした。だから，名前は見なくても読むとすぐわかったんですが。気押されるのは，自分が懸命に生きていないのを認識しているからです。認識するだけ，というのが懸命でないところです。退社される以上不可能ですが，たまにはE.O.さんの文章を読んでみたいものです。なにはともあれ，おめでとうございます。

根来　慶春(25)京都府

ほかにも，E.O.さんへのメッセージをたくさんいただきました。ありがとうございます。あの力強い文章が読めなくなったのは残念ですが，彼女はこれからも，自分の世界でがんばってくれることでしょう。

◆空きびん回収のバイトをして稼いだ金でハードディスクを買います。220時間も働いて15万円は，ちょっとキツイですね(暑かったんだよう)。

田中　正憲(18)神奈川県

時給にすると，680円くらいですか。高校生のバイト料って安いから，大変ですよね(どうしてなのかは知りませんが)。でも，自分の「労働」で欲しいものを手に入れるというのは，大切なことだと思います。ハードディスク，がんがん使っていろんなことをやってみてくださいね。

◆9月号の「数値演算の逆襲」にやられてしまいました(つまり，ついていけなかった)。高校，大学ともに商業関係の学校で，数学は俗にいう「数II」の前半まで，そのうえ，さらりと流した程度……。しかし，商業高校のなかでも理数系だった(と思っている)というプライド(？)にかけて，少しでも理解しようと思ったが，本質の部分で挫折。これを機に，数学の教科書をひっぱり出して勉強しようと思います。

大塚　正宏(19)千葉県

学校の数学の授業やテストって，なんだか生徒のランクづけのためにあるみたいで，それで数学嫌いになってしまったり，暗記だけして卒業したらすっかり忘れてしまったりする人も多いようです。でも，義務ではなく「理解したい」って気持ちから始める勉強ならば，きっと違ってくるのではないでしょうか。がんばってくださいね。

◆やっぱり会社員ともなると，余暇の時間がぐんと少なくなりますね。今では土・日が大切で大切でしかたがありません。

湯舟　幸男(21)愛知県

「時間が少ない」というのは，社会人にとっては大きな悩みですよね。でも，学生にもテストとか，受験とかの苦労があるしね。短い時間をやりくりして，本当にやりたいことをする，というのもいいことかもしれませんよ(でも，やっぱり時間はもっともっと欲しいなあ)。

◆ふと思ったのだが，西川さんが記事中で「くわしくは来月号で」と書かれたことはたいてい2カ月後になる。私が発見したので，これを「牧野の法則」とします。

P.S.僕は西川さんとZ-MUSICをいつでも応援しています。Z-MUSICがバージョン100になるまでがんばってください。牧野　裕二(19)埼玉県

応援ありがとうございます。しかし……う〜ん，2カ月後かなあ？　いえ，もしそうだとしても決してご本人のせいばかりじゃなくて，「諸般の事情」があったりするのです。ごめんしてね。

◆あれ？　表にも裏にも年齢を書くのか？　(半分ずつにしたりして！)19歳と19歳な〜んてね。

伊藤　政弘(38)愛知県

実は，両面に書いていただくとハガキの集計のときに便利なのです。読者のみなさんには申し訳ないのですが……。だから，半分になんてしないでくださいね。

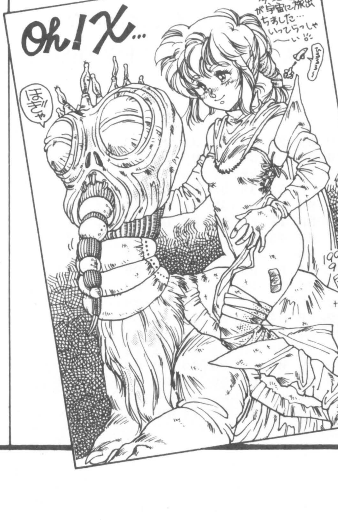
◆ボクは，学校の音楽部に所属していて，夏休み中も毎日，9月の文化祭でのオペラ「ドン・パスクワーレ」(ドニゼッティ作曲)公演のための練習に通いつめています。一応主役の独唱者なので，さぼろうものならぶり殺しものですが，楽しいので大丈夫です。音楽をすること，そしてみんなが協力することの素晴らしさを感じさせてくれる毎日です。志田　健(16)東京都

ひとりでやるのも楽しいけれど，大勢でひとつの「音楽」をつくり出す喜びはひとしおですよね。練習の厳しさとか，演奏者たちの気持ちをひとつにするための努力とかは必要ですが。この本が発売される頃には，次のどんな曲に取り組んでいるのでしょうね。

◆MATIERはとてもすごそうだ。しかし……すべての機能を使った絵って，いったいどんなものになるのだろうか。内間　正晃(20)静岡県

誰か，描いてみてくれませんか。

◆この頃，電話リクエストではなく，FAXリクエストなるものをよく見かける。いったい誰が持ってるっていうんだ，ぜいたく品だろうが，と思っておりましたが，先日，TVで耳の不自由な女の子が恋人(？)からのFAXを大事にファイルしてるのを見て，胸がつまるおもいでした。



◀鈴川　美佳子　東京都
やっぱりなんといっても，左の太モモの絆創膏がぷりてぃ！　しかし，ということは，この部分は素足なのであるな，うんうん。

◀岡村　直也　兵庫県
もはや，静かなところでは仕事がはかどらないカラダになってしまったとは。まるで，きれいに片づいてると落ち着かない，某編集部みたい……。

安藤　道子(20)宮崎県

世の中には，いろいろな立場の人がいるのですよね。そのことって，私たちはつい忘れてしまいがちですが。

◆うちの扇風機様(20歳)がコンデンサの内臓破裂を起こして倒れてしまったので，新しいのを買いにいったら，その退化ぶりに愕然としてしまった。昔の扇風機は触感バツグンのでっかいツマミとスイッチによって軽快なフットオペレーションを可能としていた。また，寝るときに暗闇の中でも操作が可能であった。ところが最近の扇風機はみーんないわゆるペコペコスイッチを採用しているため，目で見なければ操作できない。おまけに首振り角度も調整できなくなっている。リモコンをつければいいという問題ではないのだ。　石田　伯仁(19)神奈川県

確かに，目で見なければわからないスイッチって案外，不便ですよね。話はちょっと違いますが，銀行の現金引出機で，タッチパネルのものは，目の不自由な人には扱えないそうです。どういうものが「便利」なのかというのは，とても難しい問題なのかもしれません。デザインのこともありますし。でも，個人的には「足」でも操作できるようなスイッチがいいなあ。おぎょーぎの悪いワタクシとしては。

◆どなたか教えてください……。年代物の蓄音機の修理屋さんはどこ？　小型オープンリールのテープレコーダを修理してくれるのはどこ？　事業用FAX並みの図体のクセに計算機能はなんと6ケタという20年前の計算機の修理はどこ？

迫田　賢一(40)大阪府

何年か前に公開された「バカヤロー2」という映画の中でも「レコード針は手に入らない」「ビデオの規格が違ってしまって使えない」などというバカヤローの叫びがありましたが……。

◆今年の夏期バイトにより，うちのX68000は，EXPERTからXVIとクラスチェンジしようかなと思ってますが，次のアレはどーなるんだろう？

阿部　哲也(18)兵庫県

◆いま私の手にはDōGAのディスクが握られている。しかし，私はX68000を持っていない。しかし！　いまX68000を買うと，すぐに新しいX68000が出るかもしれないし……。どうすればいいんだ。ガッデム！　藍原　和久(21)大阪府

日進月歩の世になやみは尽きぬ，ですね。でも，どこかで思いきって買っちゃわないと，いつまでたっても何もできないし……。ムズカシイなあ。

◆先日ついに，念願のワープロを買った。用途は，INDEX作りがほとんど。超ぜいたくな使い方。起動はおそらく月に1回くらいでしょう。

前田　光輝(21)千葉県

「INDEX作成機」。いいかもしれない。

◆インテルのテレビCMで，「ほとんどのパソコンにはインテルが入ってる」などといっていますが，本当なのでしょうか。僕は約半分くらいだと思っていましたが。　飯田　功(17)茨城県

◀橋本　和典　東京都
移植希望をアピールするためにイラストを描いてくれました。美しいボスキャラのグラフィック，ぜひ見てみたいですね。

◀岩瀬　貴代美　福岡県
岩瀬さんは夏休みの9割を大学で過ごしてしまった，とのことですが，まさか，残りの1割はゲーセンで子供に囲まれて過ごしたとか……？

うーん。「ほとんどの」っていうのはちょっと語弊があるような気がします。たとえば「日本で使われているパソコン」とかいうように限定してしまえばもしかしたら，当たってるのかもしれませんけど。

◆私のX68000 PRO直してください。

斉藤　直史(17)山形県

編集部では，誤字・脱字とかなら直せるんですが……。

◆赤いスイッチのX68000と，青いスイッチのX68000を交換してくれる人いませんか？(おいおい)　奈良　雅雄(19)神奈川県

スイッチ交換するだけでいいのですかあ？　それなら……なんてね。

◆またもディスプレイが壊れた。前回から1カ月たっていないのに(原因が同じなら3カ月以内は無償だそうだが)。受験生でもあるまいし，なんで私が禁コンせにゃならんのだ。

佐藤　崇(24)栃木県

禁コンのあとは，その反動でガンガン使いまくり……かな？

◆ディスクが1枚死にました。初めてだったので，けっこうびっくりしました。しかし，その中身が文化祭用に作っていたプログラムだったんです。かなりな大作で，もうすぐ完成だったのに死んでしまって……。蚊取り線香をたいていたのが悪かったのでしょうかねぇ。

音羽　進(17)宮城県

まさか，ディスクの中にバグがいたので蚊取り線香にやられたのでは……？　冗談はさておき，大事なものはバックアップを取っておいたほうがいいですよ。転ばぬ先の杖，ってね。

◆僕の部屋に虫が大量発生しました。原因はOh!Xです。夏休みにマシン語をマスターしようと思った僕は，Oh!Xのバックナンバーをベッドの上に散らかしていました。すると，古い本によくいる小さくて白い虫がベッドで繁殖してしまったのです。僕は1カ月近くそのベッドで何も知らずに寝ていました。バックナンバーを持っておられる方は，たまには本の虫干しをしたほうがいいと思いますよ。西村　佳哲(22)京都府

う～ん。読み始めたとき，バグの話かと思ってドキッとしてしまいました。「原因はOh!X」なんて書いてあるから……。本当の虫でよかった，とはいいませんけど。それにしても大変でしたね。編集部でも虫干ししようかなあ。

◆寺尾響子さんも引っ越しをされたそうですが，ぼくも引っ越しました(なんのこっちゃ)。実際の「引っ越し」は1日で終わるのだけど，部屋探しから始まって新居の片づけに至るまでたっぷり2カ月忙殺され，膨大なエネルギーを使いました。おかげでDōGAお試しシステムもまだ解凍していないというありさまです。それにしても，部屋を借りる本人は認印でよいのに連帯保証人は実印を要求されるというのは変だよなあ。

河野　浩(29)東京都

編集部にも引っ越し話がもちあがっています。といっても，ちょっと南側にずれるだけなのですが。それでも，机やら本棚やらをすっかり移動するのでけっこう大変です。まずはお片づけから始めなくちゃね。さて，いったい何が「発掘」されるかな？

◆スーファミもゲームボーイも「目が悪くなる」と母に禁止されているので，父のX68000でファイナルファイトをやっています。

平野　郷(12)神奈川県

ファイナルファイトは叱られないのですか？　理由は次のうちどれでしょうか。

1)　ファイナルファイトだから。
2)　X68000だから。
3)　お父さんのパソコンだから。

◆なぜか最近充実してきたX68000。CGAやMIDI，趣味でなくても趣味になってしまう。趣味になるのはいいが，金がかかる，う～ん困ったものだ。私は現在，インドネシアに出張で来ています。このハガキも，会社のエアメールを使い，一度実家に送り返して，実家から発送しています。出張は3カ月という約束でしたが，どうも1年らしくて……。　谷口　博一(26)大阪府

インドネシアでOh!Xを読んでくださっているのですか。ありがとうございます。このハガキも海を渡って編集部に着いたとは。お体に気をつけて，仕事もがんばってくださいね。

◆いま，隣の部屋では鈴虫が鳴いています。なかなか風流なものです。ふと，「虫の鳴き声をFM音源で再現して鳴らしたら環境ソフトになるな」と思ったのでした(風鈴もいいかも)。

今井　彰彦(27)大阪府

心と体にやさしいX68000です，なんてね。今度「環境ソフト」作りに挑戦してみてくださいな。

◆(KO)さん。そのぐらいで恥ずかしがってどうするのです。まだまだ道は遠いようですね。私のように堂々と背広姿で「なかよし」を買い，自分で描いたうさぎちゃんを職場の机に飾り，みんなの尊敬を一身に集めるようにならないと，オタクの道は極められませんよ。

沼　圭司(24)静岡県

セーラームーンの単行本を買うぐらいで恥ずかしがっていては「オタク失格」なのですかあ。う～ん，オタクへの道は厳しい！(KO)さん，がんばってね（何を？）。

◆「セーラームーン」を見ている人間は僕の周りにも何人かいますが，毎週欠かさず「トマトマン」を見ているのは僕だけでしょう。主人公のトマトマンよりもニンジン忍者が好きです。

川端　洋之(21)北海道

うう，なんだか話がどんどん「オ」とか「タ」とか「ク」とかのほうへ，ずれてきてますが……。でもニンジン忍者ってかわいいのかも。見たことないけど。

◆最近，いい毒物飲料がない。チェリオには怪しい名のジュースがあるが，名前だけで味は別に怪しくない。怪しいジュースが飲みてー！

宮内　大輔(18)大阪府

あのう，怪しい「味」のジュースが飲みたいのですか？　それなら，自家製で「からしコーラ」とか「チョコレートカルピス」とかがおすすめですが，いかが？

◆6月。会社にある花壇をひとつつぶして，トウモロコシとインゲンマメと小蕪を植えた。

7月。小蕪を収穫して，漬物を作って昼に食べた。おいしかった。

8月。インゲンマメを収穫して，タマゴと一緒に炒めて食べた。おいしかった。

9月。トウモロコシが大きくなって3時のおやつに食べることができそう。楽しみ楽しみ。

しかし，我が社はいったい何なのであろう？俺の職種は試験研究員だぞ。班長ぢゃないぞ……。

原田　真志(21)静岡県

自給自足の無農薬野菜，ですか。いいなあ。でも，ほんとに会社でそんなことしていいのですか？

◆DōGAの柚姫は，茶道をやっているのですね。私も最近，会社の茶道部に入りました(会社のきれいな受付のお姉さんが茶道部の部長だから入部したなんて口が裂けてもいえません)。最近では足もしびれにくくなり，茶の湯の心がわかり始めてきました(ホンマかいな)。

横田　紀明(25)山口県

では，次のハガキでは「きれいなお姉さん」と「茶の湯の心」についてのご報告をお待ちしております。ねっ。

◆前から気になっていたのですが，「吾輩はX68000である」の記事中，右ページ右上と左ページ左下にある花のような模様はいったいなんなのですか？　泉氏の家紋なんでしょうか？それと，満開の電子ちゃんに岡村姉弟が出演していますね。見落とすところでした。

櫻井　良多郎(21)東京都

いいところに気がつかれましたね。実を申しますと，あれは，やんごとなきお生まれであります泉サマのお家(いえ)の御紋なのです。つまり，お持ち物から担当ページに至るまで，すべてあれをお入れしなければならないのです……（嘘ですが)。

◆そろそろ封印しようかな，なんて思っていたら，その前に父がキーボードをどっかに隠してしまいました。プログラムをよく作る私に，キーボードがなければ……と思っているのでしょうが，父はマウスキーボードの存在をいまだ知らない。うふふ。　幸　俊威(17)大阪府

いわれてみれば，X68000ってキーボードがなくても困らないのですねぇ。うんうん，便利，便利。でも，「うふふ」なんていってないで「お勉強」もちゃんとしてね。

◆ぼくは，物に名前をつけるのが好きだったりする。だから，ぼくの持っているX68000に似合う名前をつけてみた。その名は亀吉。ぼくはこの名前を気に入っている。きっと，ぼくのX68000も気に入っていることだろう。この亀吉と出会って1年と6カ月になるが，親にねだって大金を出して買ってもらった大切な品である。ちなみに，ぼくはあぶない人間ではありませんからよろしく！　編集部の人もパソコンに名前をつけませんか。　大島　康生(15)東京都

「おはよう，亀吉」とか「ただいま，亀吉」とかいって，かわいがっているのかな。

◆夏休みに実家の青森に帰省したときのことです。飛行機が着陸態勢に入ったのでシートベルトを締めてかまえていると，急にグィーンと上昇する感覚。しばらくすると，「青森県内の天候不良の影響で濃霧が発生し，着陸困難なため，上空で待機します」という機長のアナウンスが。結局その後も着陸を試みましたが不可能と判断し，そのまま大阪へUターン……。さすがにあのときは心身ともに参りましたよ。幸いにも，次の日の臨時便で無事に到着できました。でも，今考えると，2日間で大阪－青森間を往復半できたので得したかな，なんちゃって。

土井　準(22)大阪府

お疲れさまでした。せっかく来たのにそのまま逆戻りなんて，ちょっと悔しいですね。でも，全然関係ないところに降ろされちゃうよりはマシかも。

◆高速道路を風に吹かれながらとばすと，実に気持ちがいいものです。「まるで気分は夕涼み」というところに，気が利くのかどこからかチリーンチリーンと風鈴の音までも聞こえて……ではなくて単なるスピードの出しすぎでした。車を運転されるみなさん，くれぐれも安全運転を心がけましょう。　藤原　彰人(22)岡山県

おやおや，気をつけてくださいね。でないと……。

◆連休明けの初日の帰り（夜勤なので火曜日の朝)，朝から雨が降ったりやんだりしていた。連休中にお金を使い果たした（北海道旅行に行ってきたのだ）私はボーッとしながら家に帰ってきた。ごく普通のカーブで，スピードの出しすぎかな，と思いながらハンドルをきると，あ！曲がらない，あ～落ちる……と，田んぼにゴロン，ゴロンと車が仰向けになり，私は天井に四つん這いになっていた。あ一車が～。昨年買った車が～。借金がまだなのに一。田んぼのほうは保険ですんだのですが，車のほうは……。雨の日の運転は気をつけましょう。

赤木　英樹(24)岡山県

怪我はありませんでしたか？　雨の日は特に危ないですよね。それにしても，田んぼの賠償って，どういうのかな？　収穫できるはずだった米の代金や田んぼの修理費(？)などを見積もるのでしょうか。「破損したカカシ代」とか……。

◆以前は雑誌に掲載されているプログラムを入力すると，入力ミスでエラーが出た。現在は，入力時に最適化してしまい高速で小さいプログラムになります。ちなみにショートプロぱーてぃにあった，プロポーショナルピッチを入力し

▲玉野　健一　奈良県
あのう，ひとつだけお聞きしたいのですが，この可愛い女の子の頭にのっかってる「でろりん」ってしたものは，何……？　ちょうちょかな。

▲占部　哲彦　広島県
こちらは移植のお祝いに描いてくれました。この次は，ぜひぜひ，移植リクエストのイラストも描いてみてくださいね。

たときには，X-BASICのリストをアセンブラに直して入力し終えました。

下田　達也(25)三重県

着々とパワーアップしてるじゃないですか。私も見習わなくっちゃ。

◆ああ……宿題がぁ……順調に遅れていくぅ……誰か……助け……。　三宅　涼(14)京都府

ああ……進行がぁ……遅れ……ホンが落ち……ライターの○○さん……げんこ……。

○○さん担当(28)東京都

◆久しぶりに田舎に帰った。小さかったタマ(メス猫・1歳)もすっかり大きくなっていた。特におなかが大きかったので，アレッと思ったが，案の定，子供が入っていた。10月の試験休みに帰郷するのが非常に楽しみだ。

森上　晶仁(19)岡山県

何匹生まれるかな。お母さん似かな，それともお父さん似かな。

◆この前，大学の付属演習林へ実習に行き（私は農学部林産工学科)，生まれて初めてチェーンソーを使いました。あまりにも簡単に木がスパスパと切れてしまうのには驚きましたが，もっと驚いたのは，樹齢20年のスギの木1本の値段が4,000円程度だということでした。それでも「日本の木は高い」と，海外から木材を輸入するというのですから，日本で林業なんてやってられませんよね。　松永　正弘(22)京都府

チェーンソーって簡単なんですか？　樹木って育つのは大変なのに，あっけなく切られちゃって，しかも値段は安いなんて……。でも，もしも木材がなくなったら，すごく困りますよね，きっと。

◆なにもかもがめんどうだ～。めんどう。めんどう。どうにかしてくれ～(でも，ハガキを書くのはそれほどでもない)。

田中　信一(20)神奈川県

▲石田　伯仁　神奈川県
ワタクシの推理では，この女の子の紙袋には，たい焼きが3個入っています。おかあさんと妹へのおみやげです。根拠？　そんなのないんだけどね。

おやおや，そんなこといわないで。面倒がらずに，来月もハガキくださいね。お待ちしてます。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できないこともあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★1年前に発足した「Open Space」は，当初1年間のみの活動予定でしたが，大変好評だったため活動期間を延長することになりました。内容は，Macintoshを使ったDTP会報の発行，MacintoshとX68000のグラフィックデータ変換やソフト，アルゴリズムの開発を行っています。そのほか，小説やマンガなどなんでもありのサークルです。機種は問いません。興味のある方は，600円の無記名小為替を同封のうえ，下記の住所にお送りください。最新号をお送りします。〒399-07 長野県塩尻市片丘10391 古籏　一浩

★このたび，パソコンだけでなく「自我を文章で表現する自己確立のための思想哲学誌」をテーマとしたサークル「Wright Staff」を発足します。内容は毎月発行のペーパーメディアで，いろいろな意見を持った人の参加を待っています。興味のある方は送料実費175円分の切手を同封のうえ，下記の住所までお送りください。〒154 東京都世田谷区若林4-29-13　飯島　陸太

売ります

★X1用FM音源ボード「CZ-8BS1」を送料込みで10,000円で売ります。付属品，箱すべてあり。連絡は往復ハガキでお願いします。〒737 広島県呉市溝路町7-21　長原　泰彦

★Roland「MT-32」を32,000円で売ります。完動，目立つ傷なし。箱なし。マニュアル，ケーブル，アダプタあり。連絡は往復ハガキでお願いします。〒243-04　神奈川県海老名市国分寺台2-14-7　井上　敬介(21)

★X1用カラープロッタプリンタ「CZ-8PP2R」，RFコンバータ「CZ-8VC」，拡張I/Oポート「CZ-8BE2」，320Kバイト外部メモリ「CZ-8BE2」をまとめて20,000円くらいで売ります（すべて取扱説明書あり）。また，バラ売り可。往復ハガキに希望価格を書いてお送りください。〒658　兵庫県神戸市東灘区西岡本3-21-3　油川　正徳(24)

★Roland「CM-500」を80,000円前後，KORG「M1」を80,000円前後で売ります。「CZ-500」は箱，付属品つき。「M1」は箱なし，キャリングケース，音色カードケーブルつきです。「M1」は箱なしなので手渡し希望。まずは往復ハガキで連絡してください。〒187　東京都小平市津田町3-25-48　出羽　克康(21)

★Roland「CM-64」とカード2枚を65,000円，BOSSのモニタスピーカー「MA12AV」を2個で10,000円，X1/X68000用熱転写カラー漢字プリンタ「CZ-8PC5」を45,000円，ニューテックの100Mバイトハードディスクを50,000円以下，NECのモデム「COMSTARZ CLUB 24/5」を20,000円以下で売ります。また，「THE 福袋ver.2.0」「68000プログラマーズハンドブック」「X68000パワーアッププログラミング」「X68000ベストプログラミング入門」「X68000環境ハンドブック」「X68000マシン語入門」を適価で売ります。連絡は往復ハガキに電話番号を明記してお送りください。〒381-02 長野県上高井郡小布施町851　鈴木　理星(17)

★電子手帳用カード「ザ・ベースボール」を送料込み3,000円で売ります。付属品，箱すべてあり。連絡は往復ハガキで。〒724 広島県東広島市西条町西条東1258-5　原田　謙(18)

買います

★電子手帳とX68000の通信ケーブル「CE-300L」を送料込み1,000円で買います。付属品，箱なくてもかまいません。連絡は往復ハガキでお願いします。〒724　広島県東広島市西条町西条東1258-5　原田　謙(18)

バックナンバー

★「オールアバウトナムコ1」（電波新聞社）を送料込み4,500円＋αで買います。切り抜きさえなければ多少汚れていてもかまいません。連絡は往復ハガキでお願いします。〒369-03　埼玉県児玉郡上里町三町542　岡村　哲男(18)

★Oh!X1991年6月号を送料込み2,500円で買います。切り抜きのあるものは不可。連絡は官製ハガキでお願いします。〒573　大阪府枚方市牧野北町10-34-107　橋本　智也(16)

★Oh!MZ1984年12月号～1985年8月号，Oh!X1988年3月号，1988年6月号～1989年6月号，1989年8月号，9月号～1990年4月号を各1冊送料込み1,000円前後で買います。切り抜きは不可。そのほかのバックナンバーを売りたい方，連絡を待っています。〒167　東京都杉並区本天沼3-23-14　鈴木昇方　黒崎　亨(19)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々の意見を紹介しています。今月は9月号の内容に関するレポートです。

●9月号の目次を見ると堅そうなテーマが並んでいて，読み始めるのに少し戸惑いましたが，読んでみるとわかりやすく面白い内容でした。わかりやすいといっても，ある程度は歯応えがありました。まあ，なんの努力もなく頭にすらすら入ってくる記事よりは，そのほうがいいでしょう。数値演算プロセッサやV70ボードを私は持っていないので，それらに関する記事をつい飛ばしがちに読んでしまいましたが「そうか，こういうものなのか」とこれらの製品自体を理解するというふうに読み進めました。

矢野　啓介(19)　X68000 XVI, MZ-2500　北海道

●9月号の特集では，丹氏の最後のひと言がずしんと響きました。確かにそのとおりです。ちっとも暴言なんかじゃないです。モデリングとレンダリングの区別がなくなる，つまりは彫刻とか粘土細工をするような感覚で，CGをするってことですよね。夢のようだ……でもとても重要なことだと思います。デザイナーに優しいCGシステムを作る。それが私の理想ですが，操作性ばかりでなく速さも考えなくちゃ，と思います。やっぱりハードに頼るしかないのかしらね。あと，「MIRAGE」にはもっともっと頑張ってほしい。モデラにもまだまだ問題が残っているし，アトリビュートエディタはこれからだし，やっぱり「おおー！」といえるようになってほしい。メタボールもほしいしね。それからより速く，より表現豊かに。もちろん速いばかりではなく，半影などの多彩な表現ができるようになるといいな。どうせやるならトップを目指せ，てなもんです。

安井　百合江(18)　X68000 ACE　愛知県

●9月号の特集「数値演算の熱い逆襲」は，読み手によって評価が分かれるでしょう。まず，数学が好きか嫌いか，そして数値演算プロセッサを持っているかいないか，です。丹氏が書かれた「疑似メタボールで遊ぶ」は，いかにさぼるか，という高速化の話がメインであったため，数学嫌いの人でも楽しめたと思います。しかし，ほかの記事は数学がメインであったり，V70や数値演算プロセッサを実際に積んでいなければ，興味をひけない内容であったのではないでしょうか。特集の中では，もちろん「疑似メタボールで遊ぶ」が最高でしたね。よくぞここまでメタボールをモデル化し，高速化したものです。コンピュータ上のシミュレートには，ある程度のモデル化が必要なことはわかっていましたが，もともとコンピュータ上のものをモデル化してコンピュータ上でシミュレートする，そんなことは思いつきませんでした。これはしばらく遊べそうですね。テストが終わったらぜひとも打ち込んでみたいと思います。

高橋　毅(21)　X68000 PRO, MSX2　埼玉県

●9月号の特集の中では「FPP.MAC」が印象に残りました。数値演算プロセッサが実数演算以外に対して役に立たないことは，以前から話に聞いてましたが，いろいろ工夫をすればそれなりに使えるんだな，と思いました。FLOAT3.Xによるオーバーヘッドを減らせば，速くなるものですね。CGをやったりするならば数値演算プロセッサも役に立つでしょうが，それ以外の人にも使ってほしい，そんな気持ちが見える記事でした。また，数値演算プロセッサは素晴らしいものだ，とも思わせてくれました。

菅本　友司(21)　X68000 XVI　宮崎県

●川原氏のレビューで「MATIER」のもうひとつの顔を見たような気がしました。8月号の中野氏のレビューでは「誰でも絵が描けるツール」としての面を見せてくれました。Z's STAFFよりも安いしこれは買いやな，と思っているとさすらいの電脳絵師は「使いこなすのはたいへん」といいます。これは真の意味で「MATIER」は誰にでも受け入れられるツールではないでしょうか。とりあえず，CGを楽しみたい人には3Dペイント，細かく描きたい人にはブラシ，といった図式が私の頭の中で勝手に出来上がりました。まだ発売されていないそうですが，買ってみようと思います。自分が思っている図式を確かめるためにも……。そして，新しく始まったDōGA CGアニメーション講座は，以前の連載に比べて初心者にわかりやすくなっていると思います。結構，How To本のノリでよろしいのではないでしょうか。「DōGA CGAシステムをあのような形で配布してしまったからには，DōGAももっとしっかりしなければいけない」そんな思いがなんとなく伝わってきます。

中矢　史朗(21)　X68000 ACE-HD　愛媛県

●数値演算で特集を組んでしまう，もしくは組めてしまうのがOh!Xらしいところでしょう。興味のあるないにかかわらず，このような特集は必要だし歓迎すべきことだと思います。記事の内容はかなり高度なところもありますが，読んでいてつまらないということはありませんでした。特集の記事の中では「68881並列駆動への挑戦」がとても面白い試みだと思いました。結果的にはMPU68000自体の処理が遅すぎて，せっかくの並列化の効果が出なかった，ということでしたね。しかし，裏を返せばクロックスピードが速ければそれ相応の結果が得られる，ということなのでしょうか。文末では，超越関数で再挑戦してようやく満足のいく結果を得られたようですが，私はそこにたどり着くまでの過程が興味深かったです。

藤田　康一(21)　X68000 PRO　静岡県

ごめんなさいのコーナー

10月号　P.50　笑顔を探して

Z-MUSIC ver.1.10で演奏した場合，演奏が正常に行われないことがわかりました。リスト中のUコマンドで“－(マイナス)”の付いていないすべてのパラメータの頭に“＋(プラス)”を付け加えてください。該当箇所は，159，168，227，246，265，274，293，296，297，304，305，308行です。

10月号　P.86　ZPP.X

ZPP.Xでは，データ展開部分に16進データがある場合，A～Fの文字を音階コマンドと解釈するため正常に展開されません。データ展開部分では16進データを使わないようにしてください。

バグに関するお問い合わせは
☎03(5488)1311(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

作る楽しみ 遊ぶ楽しみ どちらがお好き？

▶特集では「ゲームマネージメント」と題して，ゲームの中核であるマネージメントシステムが，どのように成り立っているか具体的に紹介してみました。

いろいろなゲームがありますが，その根底に流れる基本的な考え方は同じです。あとはいかに作りたいものに合わせ，ゲームマネージメントシステムを発展させるだけ。特集の記事を参考にして，使い勝手のいいシステムを目指しましょう。

また，今回は，アセンブラを使った記事が多く「ゲームを作りたくても何から手をつけていいかわからない」と迷っている人には，ちょっと難しいところもあるかもしれません。そんな人は，1字1句すべてを理解しようとせず，ゲームにはどんなものが必要となるのかを探してみてください。

そして，大切なのは，思いついたことをとりあえずやってみるという姿勢です。プログラミングは，経験と勘，そして根性によって磨かれていくものですからね。自分の作りたいものを目指して努力してみましょう。身につけた技術を使い面白いものを作っていこうではありませんか。

▶来月でOh!Xは改題5周年を迎えます。Oh!Xがここまでこれたのは，ひとえに読者の皆さんのおかげ……と創刊10周年でもいったような覚えがあるセリフはおいといて（本当に感謝してます），せっかく5周年記念ですから何か面白いことをやりたいですね。すっぽんと頭のネジをすっ飛ばしてしまうような企画を立てるか，思いっきり真面目な企画で攻めるか，どちらにしても派手になることでしょう。期待していてください。

▶「X68000マシン語プログラミング」「吾輩はX68000である」「マシン語カクテルin Z80' Bar」は，著者多忙のためお休みさせていただきます。また，印刷上の不手際のため10月号の「Oh!Xreader'sぎゃらりぃ」で，伊藤浩克さんの名前が抜けていました。本当に申しわけありませんでした。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル

ソフトバンク出版部

Oh!X「㊢㊀㋮㊔」係

SHIFT・BREAK

▶誰かさんのところと違い，私の学校の研究室でX68000はコンピュータと認識されていない。某国民機はあるが，VRAMが半分だしフロッピーが外付けの「非互換機」にしかプリンタがつながっていないため不便の極致。私が使っている実験装置など総額1000万円以上もするのに，世の中の金の使い方はどこか間違っていると思うぞ，うんうん。（八）

▶昔「踊りゲーム」というものを作ろうと思っていた。キーボード全体に前転やジャンプ，スピンなどの動作を割り振っておき，入力にあわせて滑らかにキャラが踊るのだ。連係した動作も可能で，プレイヤーはより美しい踊りを摸索して遊ぶ。長い間錆びついていたアイディアだったが，ストIIの動きを見たときちょっとと思い出してしまったのよ。（哲）

▶ある晩，胸がワクワクするような面白い夢を見た。「こいつは使える！」と思って寝ぼけながらメモにストーリーを書いてまた寝た。起きて見てみると「車をカタパルトに載せて，ガス爆発の威力で塀を飛び越す。車ミサイル」と書いてある。反乱分子の役だった僕が考えた一発逆転の秘策らしいんだけど，何だかさっぱりわからない。夢ってヘンだ。（浦）

▶最近，電話をかけるのが怖い。まわりで留守電メッセージ（「名前とメッセージを入れてください，ピーッ」ってヤツ）を変にするのが流行ってるからだ。BGM付きのヤツはまだいい。サザエさんが挨拶するヤツや「もしもし。私リカちゃん。お電話ありがとー」なんてのも……。頼むから私がかけるときは留守にしないでくれっ（笑）。（で）

▶僕は対戦ポピュラスで勝ったことがない。勝負に徹しきれない性格なのか，単にヘタなのか。というわけでポピュラスIIは無念。ところでオートバイのゲームでは，国内外を問わずリアルなライディング感を実現したものが出ていない。そのヒントをつかんだような気がする。ここは理屈をこねるのでなしに実証してみたいものだ。しかし……。（A.T.）

▶ものすごい体験をしてしまった。楽器なぞできもしないのに，何人もの人と一緒に新宿ピットインのステージに立っているのである。左に渡辺香津美がいるのである。右のピアノには山下洋輔がいるのである。中央には筒井康隆がいるのである。こんなこと一生にいどどころか三生に一度あるかないかである。なんてこった。（K）

▶祝さんはカラオケが嫌いだ（電クラ53号）そうだが僕は大好きだ。仲間を募っては頻繁にカラオケに行っている。昔カラオケといえば飲み屋でやるものだったが，最近ではカラオケボックスしか考えられない。新曲の品揃えはいちばんだ。懐かしのアニメ主題歌も結構揃っていて「ソラン」や「エイトマン」を見つけたときは思わず歌ってしまった。（KO）

▶「恐龍」の本で，発見された骨の写真なんかを見ながら考えた。「もし，化石になって発見されたら」なんてね。巨大隕石かなんかが降ってきて人間が滅亡した何億年後かに，まったく違うかたちの生き物に骨を見つけられたら，……。彼らはあたしの骨から何を読み取ってくれるだろうか？　なんだか，かくれんぼみたいで，ちょっと楽しい気がした。（ふ）

▶引っ越しをした。京急の神奈川駅から徒歩8分，横浜駅まで歩いて行ける距離だから，なかなか便利な場所だ。そんな場所だから，家賃もそれなりなんだけどね。ノリと勢いで決めてしまったけど，今月の給与明細を見て……ちょっと早まったかな。ま，なんとかなるでしょう。というわけで，引っ越しの手伝いをさせられた友人O君に感謝感謝。（J）

▶VIRTUAL REALITY STUDIOを渡したときに，「やっぱりデゼニランドだな」といったことは覚えている。で，それを律儀に心に留めてオブジェクトを作ってきた（A.T.）氏。最初はなんでモヒカンの兄ちゃんを作ってきたのかわからなかったけど，なかなかやるのお。次は「やっぱり惑星メフィウスだな」といってみよう。何が出てくるか。（A）

▶これまでずいぶんいろんな特集をやってきたがゲーム特集を担当したのは実は初めてだ（どこがゲーム特集なんだという説もある）。確かに一度はやってみたかったのだが，年に数回しかない安息月を無為に消費したのではないかという気も……。Mook関係が遅れに遅れている。しかしQuadra発表から1年以上たつのになぜQuarkが走らないんだ？（U）

▶経バイト誌の広告特集（特定分野の広告を集めるために企画されたもので，広告と関連記事からなる）を開くと，どうも見覚えのある筆者名が……。ありゃま，私と同姓同名でないかい。それも「Macintoshのマルチメディア」だなんて，Oh!Xの私の立場はどうなるの。社内じゃ「バイトしたんだろう」だって。洒落にもならないよね。（T）

microOdyssey

最近の大型筐体のアーケードゲームを見てもわかるとおり，3Dゲームがポリゴナイザーなどかなり高機能なハードを引っさげ，ゲーム世界を現実世界へ近づけようとしつつある。これによって，以前に比べれば格段に表現力が上がったし，演出面でも派手に豪快になってきた。

ここでちょっと考えたいのだが，これらの技術や演出で，プレイヤーはゲーム世界へ没頭しやすくなっているのだろうか？

答えは「YES」。

技術が上がれば表現力が豊かになる，演出がよくなれば，それだけプレイヤーを引きつけることができる，という結構安易な理由だ。それなら，現在，素晴らしいゲームが生まれ続けているか，というとそうでもない。ここに，技術と演出の相関関係という問題が出てくるのだ。一例を挙げてみよう。

僕は「ギャラクシアン３」の発進シーンは，完全にレゴブロックの世界だと思った。また，艦隊戦をくぐり抜け，地上に降りて最終目的地にたどりつくと，そこに繰り広げられていたのは，ネオンサインピカピカの遊園地であった。これを見た瞬間，カッコイイと思うよりも，非常にこっぱずかしい思いを抱いたのを覚えている。まあ，こっぱずかしいまでいかなくても，違和感を覚えた人はすくなからずいるだろう。要するに完全に嘘の世界と感じた。

それとは逆に，「ギャラクシアン３」の１人用バージョンといえる「スターブレード」は，多人数バージョンより格段に機能制限があるのにかかわらず，映像の迫力は負けていない。むしろ，エンディングまでのとぎれることのない流れる映像に感動さえした。

これらの違いは映像のバランスといえる。確かにできることが多くなれば表現力が広がる。しかし，ある部分が突出すれば，全体のバランスが崩れるのは当たり前だ。そして，ゲーム世界でそのバランスが崩れたときには，無残にもただの嘘っぱちな世界となる。目立つ部分とそうでない部分のギャップに，プレイヤーがゲーム世界を現実として捕らえられなくなるのだ。そういった意味から「スターブレード」のバランスのとれた世界に，僕はよりゲーム世界の中に現実を感じることができた。

また，ゲーム画面に現れる映像は，自然映像に比べて著しく情報量が少ない。それでも僕らは，画面に展開される映像を自然に現実の映像と補完することで，リアリティを感じることができる。ただのワイヤーフレームの映像が，3D空間に浮かぶＸウイングに見えるのもそのためだ。ゲームはそこを狙ってプレイヤーをうまくだまさなくてはいけない。そして，見事プレイヤーを捕まえられたら成功，だましきれなかったらドボン。

現状では技術の先走り，という気もしないではないが，こういった問題をクリアーしてゲーム世界が構成されるとしたら……また一歩，ゲーム世界が現実に近づいていくのだろう。

そして，将来，目に飛び込んでくる映像が，自分の思い描く映像と寸分違わないもので，さらに五感をすべて再現されるものが登場したとしよう。それはゲーム世界などではなく，正真正銘自分が生きているもうひとつの現実世界となるのだ。（J）

1992年12月号11月18日(水)発売

特集　不思議なゲーム空間

・キャラクターアニメーション作法

・SX-WINDOW用おいかけっこゲーム

Oh!X 5周年記念特別企画

エレクトロニクスショウレポート

新製品紹介　X68000用版下作成支援システム

全機種共通システム

実践Small-C講座　MAKEプログラム(前編)

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートB1	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ５	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社

〒101 東京都千代田区飯田橋3-11-6

☎03(3238)0700

Oh!X 11月号

■1992年11月1日発行　定価600円(本体583円)

■発行人　孫　正義

■編集人　橋本五郎

■発売元　ソフトバンク株式会社

■出版事業部　〒108 東京都港区高輪2-19-13　NS高輪ビル

Oh!X編集部 ☎03(5488)1309

出版営業部 ☎03(5488)1360 FAX 03(5488)1364

広告営業部 ☎03(5488)1365

■印　刷　凸版印刷株式会社

©1992 SOFTBANK CORP. 雑誌 02179-11 本誌からの無断転載を禁じます。

落丁・乱丁の場合はお取り替えいたします。

今冬発売！　ユーザーのみなさん、お待たせしました。

X68000上でのプログラミング環境を充実させるために、
各種開発ツールキットをテーマ別に解説。
それが

〈 **X68k Programming Series** 〉
X68k　プログラミング　シリーズ

です。

X68k Programming Series #1 Develop.

吉野智興・中村祐一・石丸敏弘・今野幸義　共著
Ｂ5判・箱入り　2冊セット/ディスク付　予価9,000円（税込）11月上旬発売予定

本書は、X68000用に移植されているCコンパイラX68000 GCC（GCC）、アセンブラHigh speed assembler（HAS）、リンカHigh speed linker（HLK）、デバッガGNU Debugger（GDB）について新たに書き下ろしたドキュメントであり、開発キットです。付属ディスクにはこれら4種類の開発キットとサンプルプログラムが収録されています。またライブラリには、XCコンパイラおよび同シリーズの『libc』のライブラリが利用できます。
「Vol.1　Programmer's Guide」「Vol.2　Reference」の2冊より構成。Vol.1では、基礎知識やインストール方法、そしてGCC、HAS、HLK、GDBの各機能および操作方法について解説します。そしてVol.2は各種オプションスイッチやエラーの対処方法についてまとめられており、ハンディマニュアルとして最適です。

X68k Programming Series #2 libc

村上敬一郎・荻野祐二・大西恵司　共著
Ｂ5判・箱入り　2冊セット/ディスク付　予価8,500円（税込）12月中旬発売予定

本書は、Cコンパイラで利用可能なライブラリ関数を5つのレベルに分類、それぞれよく使用される約650種類のライブラリ関数を著者らが独自に開発し、その解説を書き下したものです。作成したライブラリ関数はすべて付録ディスクに収録されています。これらのライブラリ関数はX68000上で動作し、XCコンパイラおよび同シリーズの『Develop.』で使用することができます。
「Vol.1　User's Reference」「Vol.2　Programmer's Reference」の2冊より構成。Vol.1ではライブラリ関数を使用していく上での基礎知識や注意事項、およびファイル操作やユーザー管理などのライブラリ設計について解説していきます。Vol.2は付録ディスクに収録されている関数について、サンプルプログラム入りで解説したマニュアルとなっています。

SOFT BANK　ソフトバンク出版事業部

満開の電子(でんこ)ちゃん

作：船本昇竜
え：岡村 祭

講読方法：定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。

★定期購読の場合＝購読料6ヶ月分6,000円(送料サービス、消費税込)を、現金書留または郵便振替で下記の宛先へお送り下さい。

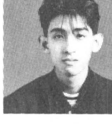
現金書留の場合：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
郵便振替の場合：東京 5－362847 ㈱満開製作所

- ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
- 3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
- 新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
- 製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。

★TAKERUでお求めの場合＝1部につき1,200円(消費税込)です。

- 定期購読版と内容が一部異なる場合があります。御了承下さい。
- お問い合わせ先 TEL(03)3554－9282(月～金 午前11時～午後6時)

(なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります)



船越 直弥（北海道）

私は、X68Kを持っていないのですが、電脳倶楽部は創刊以来あこがれでした。そこにX68Kを購入した後輩が出来たのです。あー、神様、祝様、これで、あこがれの電倶が楽しめる。しかし、そこには落し穴が…。

函館には当時タケルがなかったのでした。こんなことだったら、定期購読しておくんだったと後悔し、結局、嫌がる後輩をつきあわせて、わざわざ札幌のタケルまで買いにいったのでした。

―教訓―

「定期購読、家で待ってりゃ電源オンですぐ起動」

絶賛発売中！
CD-ROM Drive for X68000
マルチメディアへの誘い

FirstClassTechnology制作のCD-ROM Device Driverを付属させ、ついにX68000用CD-ROM Driveの登場です。本製品を使用することにより、MS-DOSやPC-9801シリーズ、FM-TOWNSなどで採用されている、ISO9660規格のCDをHuman68K/SX-WINDOWで直接扱えるようになります。
　また、将来の拡張にも柔軟に対応できるSCSIインターフェースによる接続を採用。ディジーチェーンによって既存のSCSIハードディスクとの同時使用も可能です。

ドライブ仕様

型番	KGU-XCD
使用ドライブ	東芝　XM-3301
平均アクセスタイム	325mSEC
インターフェース	SCSI
キャッシュメモリー	64KB
オーデオ出力	RCA-Phono端子×2 ステレオヘッドホン端子
電源	専用ACアダプター
外形寸法	150×228×50（電源部含まず）

※SCSIケーブル・ターミネーターは別売になります。

付属サポートソフト

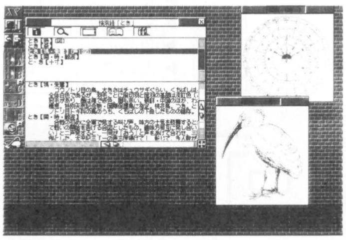
ISO9660準拠デバイスドライバ
MusicPlayer for SX-Window
Macintosh™用ファイルビューア　for SX-Window

KGU-XCD対応
X68000 CD-ROM第一弾！「フリーウェア集」
Free Soft Ware Selection - CD68K
近日発売

標準価格￥118,000-

X68000 Pro SHOP
BASIC HOUSE
KEISOKUGIKEN Corp.
TEL 0286-22-9811　FAX 0286-25-3970

CD-ROM広辞苑 検索&閲覧
SX-広辞苑



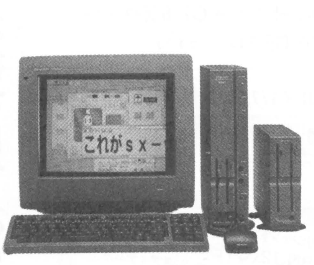
富士通・NECより発売されている12cmCD-ROM広辞苑を検索/参照し、X68000上で使用できるようにする専用検索ソフトです。通常の検索に加えて、文章中語検索、条件検索などの強力な検索機能が利用でき、広辞苑内に収録されている図版の表示、音声の再生も可能です。SX-WINDOW上で動作するので、SX-WINDOWの特徴である疑似マルチタスク機能やカット＆ペースト等の機能が利用でき、エディタXなどで文章編集中に検索/参照することが簡単にできます。また将来ワープロなどのSX-WINDOW対応ソフトが発売された場合にも本ソフトの使用によって広辞苑の有効利用が可能になります。同時に複数起動ができるため、いくつかの情報を同時に参照しながら作業を進めることも可能です。

標準価格　検索ソフトのみ　￥19,800-
　　　　　CD広辞苑セット　￥47,800-

※本アプリケーションの実行にはメインメモリ2Mバイト以上実装している必要があります。

通　販　大　特　価

X68000 Compact XVI

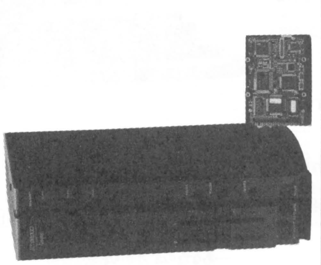


CZ-606D/CZ-6FD5 SET
定価合計￥477,600
特価￥338,000

X68000 Compact XVI

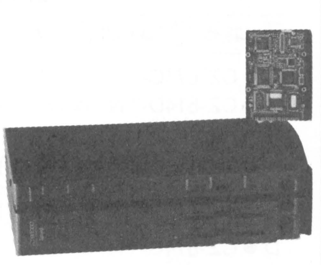
CZ-606D/CZ-8PC5-BK SET
定価合計￥474,600
特価￥358,000

X68000 Compact HD80



2.5″80MBHD/CZ-606D SET
定価合計￥545,800
特価￥399,800

X68000 Compact HD120



2.5″120MBHD/CZ-606D SET
定価合計￥575,800
特価￥428,000

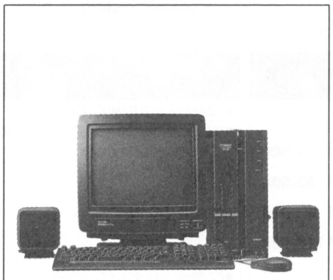
X68000 XVI HD100

CZ-606D/Quantum100SET
定価合計￥565,800
特価￥398,000

X68000 XVI HD120

CZ-606D/Quantum120SET
定価合計￥575,800
特価￥408,000

X68000 XVI HD240



CZ-606D/Quantum240SET
定価合計￥625,800
特価￥438,000

X68000 XVI HD425

CZ-606D/Quantum425SET
定価合計￥745,800
特価￥528,000

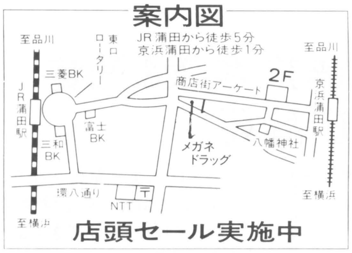
低金利クレジット　通信販売送料 全国一律￥1,000 長期クレジット可能
※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップ BASIC HOUSE
本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811　FAX 0286-25-3970

■店頭にて、新作ゲームソフト25～30%OFF!!（税別）、超低金利オクトハッピークレジットをご利用下さい!!

■冬のボーナス一括〈10・11・12月末〉払いOK!!手数料無料!!ご利用下さい。

パソコンプラザ OCT オクト



——オクトで始まるパソコンワールド——

☎03-3730-6271

●営業時間 AM 11:00～9:00/日曜・祭日PM7:00　電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX03-3730-6273

全国通販

定休日:毎週火曜日
（祭日の場合翌日になります。）

OCT-1 システム インフォメーション

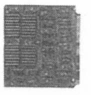
- ▶全商品保証付（メーカー保証）
- ▶超低金利ハッピークレジット（1回～60回）頭金ナシOK!
- ▶ボーナス一括払いOK! ボーナス 2回・4回・6回 払いOK!
- ▶配達日の指定OK!（万全なサポート体制）
- ▶商品の組合せ自由! オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

特選周辺機器（送料￥500）

■SX-68MⅡ MIDIインターフェイスボード
（システムサコム）￥19,800… 特価￥13,500

■モデム〔FMMD-311G〕・2400bps.クラス5
（Fujitsu）￥35,800……… 特価￥24,000

■増設RAMボード＝I・Oデータ



①PIO-6BE1-A（1MB）
￥25,000………… 特価￥15,800
②PIO-6BE2-2M（2MB）
￥50,000………… 特価￥31,000
③PIO-6BE4-4M（4MB）
￥88,000………… 特価￥54,000
④SH-6BE1-1M（1MB）
￥25,000………… 特価￥17,800

■便利です。夜9時まで営業しております。お立寄り下さい。お待ちしております!!

OCT-1 蒲田

皆様方へ、残暑厳しい折の為!!
残暑お見舞いセール! 実施中!!

SHARP

X68000 Compact
PERSONAL WORKSTATION・XVI

- 16MHz
- SX-WINDOW ver1.1
- Attachment MEMORY BORD

■CZ-674C-TN（定価￥298,000）

Ⓐ ●CZ-674C-TN
●CZ-608D-TN（14型カラーディスプレイ）

定価合計￥392,800▶**超特価￥283,000**

12回	￥25,900	24回	￥13,700	36回	￥ 9,500	48回	￥ 7,500

Ⓑ ●CZ-674C-TN
●CZ-607D-TN（14型カラーディスプレイTV）

定価合計￥397,800▶**超特価￥285,000**

12回	￥26,100	24回	￥13,800	36回	￥ 9,600	48回	￥ 7,500

Ⓒ ●CZ-674C-TN
●CZ-614D-TN（15型カラーディスプレイTV）

定価合計￥433,000▶**超特価￥310,000**

12回	￥28,300	24回	￥15,000	36回	￥10,400	48回	￥ 8,200

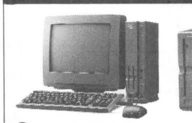
Ⓓ ●CZ-674C-TN
●CZ-606D-TN（14型カラーディスプレイ）

定価合計￥377,800▶**超特価￥270,000**

12回	￥24,700	24回	￥13,100	36回	￥ 9,100	48回	￥ 7,100

※送料￥2,000・税別
（クレジット価格は、送料・税込）

Compact限定セット



【送料￥3,000・税別 クレジットは送料・税込】

①CZ-674CH〈本体〉＋CZ-608DH〈モニター〉
＋CZ-6FD5（5インチドライブ）

定価合計￥492,600▶**超特価￥320,000**

12回	￥29,300	24回	￥15,500	36回	￥10,800	48回	￥ 8,500

②CZ-674CH〈本体〉＋CZ-606D〈モニター〉
＋CZ-6FD5（5インチドライブ）

定価合計￥477,600▶**超特価￥310,000**

12回	￥28,400	24回	￥15,100	36回	￥10,400	48回	￥ 8,200

X68000 Compact 大好評記念プレゼント!!
—あなたのオクトから素敵な贈物—

今、Compactをお買い上げいただいた方は、プレゼントの①番か②番のどちらかをお選び下さい。プラス③番は、もれなくプレゼント!!

①

or

②

※ソフトは変更出来ます!

③ ●MF-2HD（5枚）
●シリコンキーボードカバー
もれなく!! サービス!!

※どちらかお選び下さい!!
（どっちが得かヨーク考えてネ!）

ハードディスク（送料￥1,000）

■システムサコム SCSI
- ●HD-J040
￥ 89,000 42M/25ms……… 大特価￥ 61,000
- ●HD-J100
￥128,000 100M/20ms…… 大特価￥ 83,000
- ●HD-J130
￥148,000 130M/20ms…… 大特価￥ 99,000
- ●HD-J170
￥189,800 173M/20ms…… 大特価￥116,000

■ロジテック SCSI
- ●LHD-FM100E
￥ 99,800 100M…………… 大特価￥ 69,000
- ●LHD-FM200E
￥138,000 200M/17ms…… 大特価￥ 95,000

■エニックス SCSI
- ●EFX-100B
￥118,000 100M/19ms …大特価TEL下さい
- ●EFX-140B
￥138,000 140M/16ms …大特価TEL下さい

周辺機器コーナー（送料￥500）

品名	定価	特価
●CZ-6BE2A 2MB RAM（CZ-634C/644C用）	（￥ 59,800）	▶特価￥42,500
●CZ-6BE2B 2MB RAM（CZ-634C/644C/674C用）	（￥ 54,800）	▶特価￥39,200
●CZ-6BE2D 2MB RAM（CZ-674C用）	（￥ 54,800）	▶特価￥39,000
●CZ-6BE2 2MB RAM	（￥ 79,800）	▶特価￥59,000
●CZ-6BE4C 4MB RAM	（￥ 98,000）	▶特価￥73,000
●CZ-6BFI 増設用RS-232ボード	（￥ 49,800）	▶特価￥35,800
●CZ-6BGI GP-IBボード	（￥ 59,800）	▶特価￥42,800
●CZ-6BMI MIDIボード	（￥ 26,800）	▶特価￥19,200
●CZ-6BNI スキャナ用パラレルボード	（￥ 29,800）	▶特価￥21,500
●CZ-6BPI 数値演算プロセッサボード	（￥ 79,800）	▶特価￥57,000
●CZ-6BOI ユニバーサルI/Oボード	（￥ 39,800）	▶特価￥29,800
●CZ-6CRI CZ-6TU用RGBケーブル	（￥ 4,500）	▶特価￥ 3,400
●CZ-6VTI/BK カラーイメージ・ユニット	（￥ 69,800）	▶特価￥52,000
●CZ-8NM2A マウス	（￥ 6,800）	▶特価￥ 5,100
●CZ-8NTI マウストラックボール	（￥ 9,800）	▶特価￥ 7,300
●CZ-8NSI カラーイメージスキャナ	（￥188,000）	▶特価￥132,000
●CZ-6BCI FAXボード	（￥ 79,800）	▶特価￥ 57,000
●CZ-8TM2 モデムユニット	（￥ 49,800）	▶特価￥ 37,000
●LC-10CI-H カラー液晶ディスプレイ	（￥ 59,800）	▶特価￥ 45,800
●CZ-6TU GY/BK RGBシステムチューナー	（￥ 33,100）	▶特価￥ 23,800
●BF-68PRO 高性能CRTフィルター	（￥ 19,800）	▶特価￥ 14,500
●CZ-6MOI 光磁気ディスクユニット	（￥450,000）	▶特価￥330,000
●CZ-6BSI SCSIインターフェースボード	（￥ 29,800）	▶特価￥ 22,000
●CZ-6BL2 LANボード	（￥298,800）	▶特価￥219,000
●CZ-6BVI（ビデオボード）	（￥ 21,000）	▶特価￥ 15,400
●CZ-6BP2 数値演算プロセッサ	（￥ 45,800）	▶特価￥ 34,300
●AN-S100 スピーカーシステム（2本1組）	（￥ 36,600）	▶特価￥ 26,300
●JX-220X カラーイメージスキャナー	（￥168,000）	▶特価￥120,000
●CZ-6CSI SCSI変換ケーブル	（￥ 12,000）	▶特価￥ 8,900
●CZ-6CTI CZ-6TU用テレビコントロールケーブル	（￥ 5,500）	▶特価￥ 4,200

※クレジットの回数は1回～60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料無料（注）本体セット以外の周辺機器（プリンター、モデム、HDD等）及びソフトの送料は、北海道・九州地区＝1ケロ￥1500、■その他離島地区は、1ケロ￥2000となります。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい!!

■ビッグバーゲンセール実施中!!ゲームソフト(ビジネス)新製品続々入荷中!!

■ボーナス〈2回・4回・6回〉払いOK!!手数料無料!!ご利用下さい。■店頭にて、新作ゲームソフト25〜30%OFF!!

X68000 ——堂々のラインアップ!!——

■PROII/XVI■

ボーナス〈2回・4回・6回〉払いOK!!手数料無料!!

X68000 XVI エクシヴィ

(送料・消費税込)

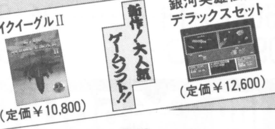
超特価¥残念!表示不能!!

X68000XVI ドッカーン!プレゼント!!

—あなたのオクトから素敵な贈物—

今、XVIをお買い上げいただいた方は、プレゼントの①番か②番のどちらかお選び下さい。プラス③番はもれなくプレゼント!!

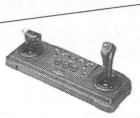
① F15ストライクイーグルII (定価¥10,800)　新作!大人気ゲームソフト!!　銀河英雄伝説II デラックスセット (定価¥12,600)



or

※どちらかお選び下さい!!

② インテリジェントコントローラ ■CZ-8NJ2 (CYBER STICK) シューティングゲーマーの必須アイテム!! (定価¥23,800)



+

③ MD-2HD(10枚)／シリコンキーボードカバー　もれなく!!サービス!!

■CZ-634C-TN (定価¥368,000)

Ⓐ●CZ-634C-TN+CZ-606D-TN
定価合計¥447,800▶超特価¥表示不能!

12回	24回	36回	48回
~~¥27,600~~	~~¥14,600~~	~~¥10,100~~	~~¥8,000~~

Ⓑ●CZ-634C-TN+CZ-614D-TN
定価合計¥503,000▶超特価¥表示不能!

12回	24回	36回	48回
~~¥31,200~~	~~¥16,600~~	~~¥11,500~~	~~¥9,000~~

■CZ-644C-TN (定価¥518,000)

Ⓒ●CZ-644C-TN+CZ-606D-TN
定価合計¥597,800▶超特価¥表示不能!

12回	24回	36回	48回
~~¥37,400~~	~~¥19,800~~	~~¥13,700~~	~~¥10,800~~

Ⓓ●CZ-644C-TN+CZ-614D-TN
定価合計¥653,000▶超特価¥表示不能!

12回	24回	36回	48回
~~¥40,900~~	~~¥21,700~~	~~¥15,000~~	~~¥11,800~~

※クレジット表は、送料・消費税込!

X68000PROII

ラストチャンス!!
〈BIGプレゼント付〉

(送料無料・税別)

X68000PROII
(CZ-653C)

定価¥285,000
超特価~~¥138,000~~

新作!大人気ゲームソフト!!　銀河英雄伝説II デラックスセット (定価¥10,800)　プレゼント

さらに!　★JOY CARD(連射式)×2個
さらにさらに!!　★MD-2HD 10枚

■CZ-653C (定価¥285,000)

Ⓐ●CZ-653C+CU-21HD
定価合計¥433,000▶超特価¥239,000

12回	24回	36回	48回
¥22,100	¥11,700	¥8,100	¥6,400

Ⓑ●CZ-653C+CZ-606D
定価合計¥364,800▶超特価¥195,000

12回	24回	36回	48回
¥18,100	¥9,600	¥6,600	¥5,200

Ⓒ●CZ-653C+CZ-607D
定価合計¥384,800▶超特価¥209,000

12回	24回	36回	48回
¥19,400	¥10,300	¥7,100	¥5,600

Ⓓ●CZ-653C+CZ-614D
定価合計¥420,000▶超特価¥229,000

12回	24回	36回	48回
¥21,200	¥11,200	¥7,800	¥6,100

X68000ソフト大セール実施中!!(ゲームソフト25〜30%OFF)　(送料¥500)

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0 (シャフト)定価¥58,000 ……特価¥36,500

〈開発ツール〉●CコンパイラPRO-68K Ver.2.1 定価¥44,800 CZ-285LSD ……特価¥32,500

〈統合表計算ソフト〉BUSINESS PRO-68K Popular 定価¥28,000 CZ-286BSD ……特価¥21,000

〈レイアウト〉●Pressconductor PRO-68K 定価¥28,000 CZ-268BSD ……特価¥21,000

〈C言語〉●C & Professional Pack 定価¥58,000 ……特価¥39,600

〈音楽〉●Music studio PRO-68K Ver. 2.0 定価¥28,800 CZ-261MS ……特価¥21,200

〈CGシール〉●CANVAS PRO-68K 定価¥29,800 CZ-249GS ……特価¥22,200

〈ワープロ〉●Multiword Ver.1.1 定価¥32,000 CZ-225BSD ……特価¥23,000

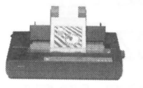
〈OS〉●OS-9 X68000 Ver.2.4 定価¥35,800 CZ-284SSD ……特価¥26,900

型名	商品	定価	特価
CZ-212BS	〈BUSINESS PRO-68K〉	(¥ 68,000)	¥ 48,000
CZ-213MS	〈MUSIC PRO68K〉	(¥ 18,800)	¥ 13,200
CZ-275MWD	〈SOUND SX-68K〉	(¥)	¥TEL下さい
CZ-215MS	〈Sampling PRO-68K〉	(¥ 17,800)	¥ 12,500
CZ-287SS	〈SX-WINDOW Ver.2.0〉	(¥ 12,800)	¥ 9,600
CZ-220BS	〈DATA PRO-68K〉	(¥ 58,000)	¥ 40,000
CZ-272CWD	〈Communication SX-68K〉	(¥ 19,800)	¥ 15,300
CZ-224LS	〈THE 福袋 V2.0〉	(¥ 9,900)	¥ 7,400
CZ-253BS	〈CARD PRO-68K Ver.2.0〉	(¥ 29,800)	¥ 20,800
CZ-258BS	〈Tlepotion PRO-68K〉	(¥ 22,800)	¥ 16,800
CZ-244SS	〈Homan 68K Ver.2.0〉	(¥ 9,800)	¥ 7,500
CZ-247MS	〈MUSIC PRO-68K〔MIDI〕〉	(¥ 28,800)	¥ 20,800
CZ-240BS	〈Stationery PRO-68K〉	(¥ 14,800)	¥ 11,500
CZ-243BS	〈CYBER NOTE PRO-68K〉	(¥ 19,800)	¥ 15,200

型名	商品	定価	特価
	〈Z's TRIPHY(デジタルクラフト)〉	(¥ 39,800)	¥ 27,000
	〈テラッツオ(ハミングバード)〉	(¥ 19,400)	¥ 13,600
	〈KAMIKAZE(サムシンググッド)〉	(¥ 68,000)	¥ 43,800
	〈Final Ver.3.2(エーエスピー)〉	(¥ 38,000)	¥ 29,000
	〈サイクロンEXPRESSα68〉	(¥ 98,000)	¥ 69,000
	〈Gツール(ザインソフト)〉	(¥ 28,000)	¥ 18,600
	〈たーみのる2(SPS)〉	(¥ 17,800)	¥ 13,000
	〈G68K Ver.2 PRO〉	(¥ 22,000)	¥ 17,300
	〈C-TRACE68 Ver.3.0〉	(¥ 98,000)	¥ 68,500
CZ-251BS	〈ハイパーワード〉	(¥ 39,800)	¥ 29,400
CZ-260LS	〈XBAS to CHECKER PRO-68K〉	(¥ 9.800)	¥ 7,500
CZ-234LS	〈AI-68K〉	(¥188,000)	¥139,000
CZ-255GS	〈CANVASドローグラフィックLIB〉	(¥ 8,800)	¥ 6,600
CZ-256GS	〈CANVASドローグラフィックVol.2〉	(¥ 8,800)	¥ 6,600

プリンタ　(送料¥1,000)

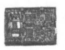
■CZ-8PC5-BK
熱転写カラー漢字
定価¥96,800
大特価¥68,800



■IO-735X-B
カラーイメージジェット
定価¥248,000
大特価¥154,000

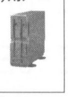
今月の推奨品　(送料¥1,000)

■内蔵用ハードディスク
〈Compact XVI(CZ-674C)用〉
〔KGU-HD80K〕
Compact HD-80キット
定価¥168,000



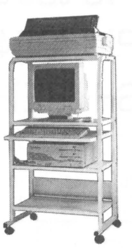
限定特別価格¥TEL下さい!

■5インチフロッピーディスクユニット
〈X68000Compact(CZ-674C-H)用〉
〔CZ-6FD5〕
定価¥99,800

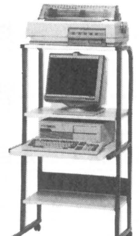


限定特別価格¥TEL下さい!

パソコンラック〈送料無料〉



Ⓐ5段キャスター付
スライド式キーボード台
●1150(H)×640(W)×600(D)
定価¥38,000
特価 ¥12,500



Ⓑ4段キャスター付
●1250(H)×640(W)×700(D)
定価¥29,800
特価 ¥8,800

店頭新作ゲームソフト25〜30%OFF!!ビジネスソフト25%より特価中

★通信販売お申込みのご案内★　〒144 東京都大田区蒲田4-6-7 TEL:03-3730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

回数	%	回数	%	回数	%	回数	%
3回	3.5	6回	4.5	10回	6.0	12回	6.0
15回	9.0	18回	11.0	20回	12.0	24回	12.5
30回	17.0	36回	17.5	48回	23.0	60回	33.0

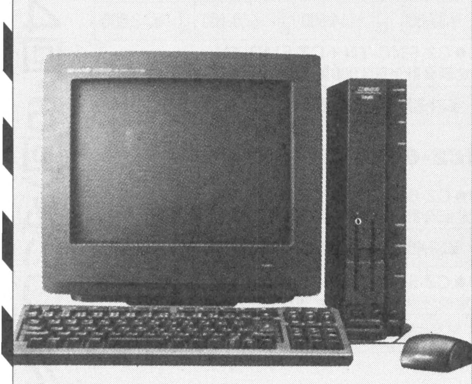
振込先
富士銀行 久ヶ原(クガハラ)支店 当No.1824
三菱銀行 蒲田支店 当No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

マイコンショップ川口

☎0482-25-1718

（消費税別）



New X68000
COMPACT XVI
~~¥298,000~~

CZ-674C-H……………¥298,000
CZ-608D-H……………¥ 94,800
AV-090-SC……………¥168,000

定価 ¥560,000
超 特 価

ソフト各種超特価ご奉仕中

CZ-219SS OS-9/X68000……定価¥29,800
CZ-213MS MUSIC PRO68K 定価¥18,800
CZ-214MS SOUND PRO68K 定価¥15,800
CZ-215MS Sampling PRO68K………定価¥17,800
CZ-220BS DATA PRO68K…定価¥58,000
CZ-224LS The 福袋 Ver2.0…定価¥ 9,980
CZ-225BS Multiword………定価¥32,000
CZ-251BS Hyper word……定価¥39,800

中古売買価格表

品　名	買取り価格	売　価
CZ-633C	160,000より	180,000より
CZ-644C	210,000より	230,000より
CZ-613C	105,000より	125,000より
CZ-603C	75,000より	95,000より
CZ-612C	85,000より	98,000より
CZ-602C	65,000より	85,000より
CZ-653C	75,000より	95,000より
CZ-663C	95,000より	115,000より
CZ-662C	75,000より	98,000より
CZ-652C	55,000より	75,000より
CZ-611C	70,000より	89,000より
CZ-601C	45,000より	65,000より
CZ-612D	35,000より	45,000より
CZ-602D	30,000より	39,800より
CZ-603D	20,000より	29,800より
CZ-604D	25,000より	34,800より
CZ-605D	45,000より	55,000より

プリンター

CZ-6VT1……………特価¥ 47,700
CZ-8PG1……………特価¥ 86,800
CZ-8PG2……………特価¥106,900
CZ-8PK10……………特価¥ 66,800
CZ-8NS1……………特価¥141,000
CZ-6BC1……………特価¥
CZ-6BG1……………特価¥
CZ-6BP1……………特価¥
CZ-6BP2……………特価¥ 34,400

ラムボード

CZ-6BE2A…定価¥59,800…特価¥ 44,900
CZ-6BE2B…定価¥54,800…特価¥ 41,100
CZ-6BE2D…定価¥54,800 特価¥ 41,100
CZ-6BE1B…定価¥28,000…特価¥ 21,000
CZ-6BE2……定価¥79,800…特価¥
CZ-6BE4C …定価¥98,000…特価¥
PIO-6BE1-A…定価¥25,000…特価¥
PIO-6BE2-2M 定価¥50,000…特価¥
PIO-6BE4-4M 定価¥88,000…特価¥
SH-6BE1-1M…定価¥25,000…特価¥

ファイル

CZ-6MO1……………定価¥450,000 特価¥
CZ-64H……………定価¥120,000 特価¥
CZ-68H……………定価¥160,000 特価¥

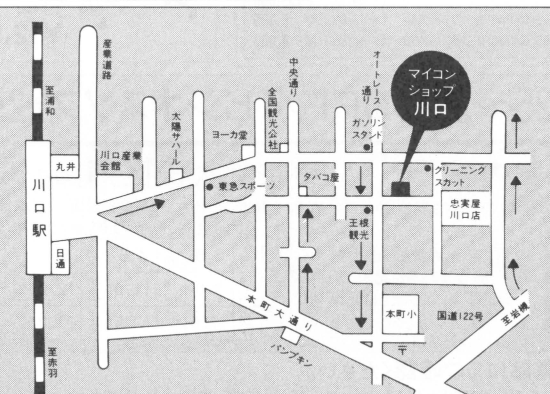
その他機種

CZ-8NS1 カラーイメージスキャナ‥定価¥188,000 特価¥
JX-220X カラーイメージスキャナ…定価¥168,000 特価¥
CZ-6BN1 スキャナ用パラレルボード‥定価¥ 29,800 特価¥
CZ-6VT1 カラーイメージユニット‥定価¥ 69,800 特価¥
CZ-6BV1 ビデオボード ………定価¥ 21,000 特価¥
CZ-8TM2 モデムユニット………定価¥ 49,800 特価¥
CZ-8NJ2 インテリジェントコントローラ………定価¥ 23,800 特価¥
CZ-8NM3 マウス・トラックボール‥定価¥ 9,800 特価¥
CZ-8NT1 トラックボール………定価¥ 6,888 特価¥
CZ-8NJ1 ジョイカード…………定価¥ 1,700 特価¥
CZ-6BC1 FAXボード…………定価¥ 79,800 特価¥
CZ-6BM1A MIDIボード ………定価¥ 26,800 特価¥
CZ-6BP1 数値演算プロセッサ……定価¥ 79,800 特価¥
CZ-6BP2 数値演算プロセッサ……定価¥ 45,800 特価¥
CZ-6TU-BK-GY RGBシステムチューナー 定価¥ 33,100 特価¥

★クレジット回数1～60回まで設定自由

回　数	1	3	6	12	15	20	24	36	42	48	54	60
金利（%）	2.5	3.5	4.5	6	9	12	12.5	17.5	22	23	28.5	29.5

中古品も取扱っております。



矢印の道路は一方通行です。

通信販売をご利用の方

—全国通販—

通信販売をご利用の方は、売値の変動がありますので在庫、値段をあらかじめ確認のうえ電話で、商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。

X68000シリーズ 大特価セール！

10月末迄

AIBIT
アイビット電子株式会社

30%〜50%OFF
店舗移転・在庫処分大特価
本体・ディスプレイ・プリンター・ソフト
（シャープ商品及び関連商品）

※特価リスト差し上げます。ハガキにて申し込み下さい。

他周辺機器及びポケコン全機種取り扱い。

〈全商品新品完全保証付〉
★シャープ・シャープ周辺機器（拡張機器全機種、プリンター他）・富士通・NEC常時取り扱い。
★シャープ・カシオポケコン全機種取り扱い。PACIFIC・YHP・キャノンも取り扱い。
★学校、企業納入受け賜ります。送料別料金。★上記商品価格には、消費税は含まれておりません。
★特価表及び資料をご希望の方は、72円切手を同封の上お送りください。

通信販売のお問い合せ、御注文は
TEL.0426-45-3001(本店) FAX.0426-44-6002
●営業時間/10:00〜19:00●電話受付/9:00〜22:00迄可●定休日/水曜日
SHARP SUPER EXE SHOP
アイビット電子株式会社 〒192 東京都八王子市北野町560-5

上記の広告商品はすべて店頭販売もしております。

全国通信販売
北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店 （普）1752505

SHARP

コンピューター事業拡張につき
プログラマー募集！

提供するのは、X68000の才能をひき出す仕事です。

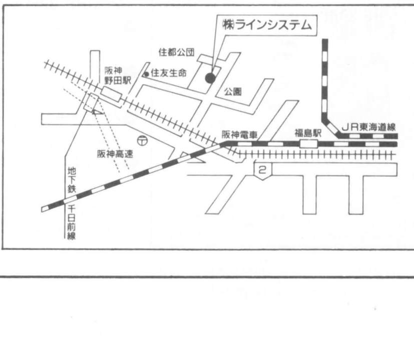
勤務地 大阪・東京・岡山
（男女不問・現地面接可）

■会社概要
設　　立■昭和44年
資 本 金■1,500万円
従業員数■25名
平均年齢■26歳
■事業内容
パーソナルコンピュータ・AXによる自社ソフトパッケージの開発及びオーダーメイド販売サポート
X68000による画像作成業務
資　　格■高卒以上30歳位迄の方
※C言語、アッセンブラーの出来る方歓迎。未経験者も歓迎。
給　　与■経験・能力等考慮の上、当社規定により優遇いたします。例 25歳　㊊ 176,000円
※別途報奨金制度あり
待　　遇■昇給年1回・賞与年2回 手当/業務・営業・皆勤 交通費全額支給
勤務時間■9:00〜18:00
福利厚生■各種社会保険完備 退職金制度 財形貯蓄制度 社内旅行有

経験の有無を問わず、X68000大好き人間 歓迎。経験者には、実力を発揮する場を、未経験者には丁寧な指導をお約束します。
シャープ、XEROX等のシステム機器販売から、シャープ・コンピューターのシステムプレゼンテーターとしてメーカーの期待を担う当社で活躍して下さい。

株式会社ラインシステム
本社 〒553 大阪市福島区鷺洲3丁目1 TEL06-458-7313 担当 菊田
〒115 東京都北区浮間3-2-16 エスポワール403 TEL03-5994-2087

休日休暇■隔週休2日制（完全週休2日制も検討中）
祝日
有給・特別・夏期・年末年始休暇等
応　　募■履歴書（写真貼付）を持参又は本社まで郵送して下さい。追って詳細を連絡致します。関東方面での面接に関しては本社からの連絡後、東京事務所にて行います。
※入社日相談に応じます。
※応募の秘密厳守いたします。
交　　通■阪神、地下鉄野田駅下車 徒歩7分



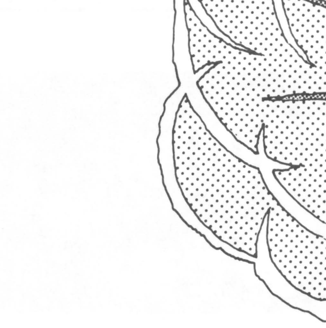
ACCESS
for X68000

V70アクセラレータがどうして速いのか
そなただけにそっと教えてしんぜよう。
実はな、魔法のランプのように
魔人が住んでいるからなのじゃよ。
この魔人は、X68000の命令なら
何でも言う事を聞くのじゃが
特に芸術的な事ならなんでもござれの
凄いやつじゃ。
グラフィック演算でも、レイトレーシングでも
そなたが命令すればあっという間じゃ！
V70とX68000との面倒なやりとりも
全部やってくれるから
そなたはゆっくり茶でも飲んでいてくだされ。
おっと、そろそろ絵が描き上がる頃じゃ。
このことは決して他の者に教えるでないぞ……。

誰も知らなかった

V70のヒ・ミ・ツ

V70 アクセラレータ　VDTK-X68K



※製作：ボード……………有限会社アクセス
ソフトウェア………株式会社ハドソン

VDTK-X68Kの仕様

- ●V70 CPU(μPD70632)
  20MHz 32ビットマイクロプロセッサ
- ●V70AFPP(μPD72691)
  フローティング・ポイント・プロセッサ
- ●メインメモリ(DRAM)2Mバイト
  同一ページ内のアクセスはNo Wait
- ●共有メモリ(SRAM)128Kバイト
  X68000との通信用
- ●併行動作　X68000とV70は、併行に動作することが可能。
  データの受け渡し処理のために双方向ハンドシェークI/Oポートを搭載。

同梱ソフトウェア

- ●アセンブラ
- ●リンカ
- ●ソースコードデバッガ
- ●システムモニタ
- ●フロートエミュレータ
- ●コマンドシェル

オプションソフトウェア

- ●Cコンパイラ
  (VDTK-C-X68K)

価格

●ボードパッケージ　(XVI対応)
VDTK-X68K……………………¥248,000

●オプションソフト(Cコンパイラ)
VDTK-C-X68K…………………¥68,000

購入方法

上記商品は当面の間、通信販売のみとさせて頂きます。
購入ご希望の方は、住所、(社名、所属)氏名、電話番号をお知らせ下さい。注文書をお送りいたします。

有限会社アクセス　〒101 東京都千代田区神田神保町1-64 神保町協和ビル7F
☎03(3233)0200㈹　FAX.03(3291)7019

資料請求券
oh!X
11月号

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

タイムトラベルシリーズ

第8回

《もし、この時代にパソコン通信があったなら》

こっちはどっちだ！大冒険。

不思議の国のアリス

川べりでうとうとしていたアリスが見つけたのは、チョッキから時計を取り出して見ているウサギでした。ウサギを追って入った穴の中は、奇妙で不思議な、冒険の世界。体の大きさが変ったり、ニヤニヤ笑うチェシャ猫に出会ったり。いつもお茶ばかり飲む帽子屋や、なげいてばかりいるニセウミガメたちと、お話とも言葉遊びともつかない時間を過ごしていきます。そうこうするうち、トランプ王国の裁判にひっぱり出され、証言台に。一同が集まるその席上で、アリスはまたまた体が大きくなって……。気がつくと、川べり。お姉さんのひざの上でアリスは夢からさめたのでした。

もし、この時代にパソコン通信があったなら。

もし、この時代にパソコン通信があったなら、アリスの冒険はもっと波乱万丈になっていたでしょう。アリスはいつも不思議な人に出会う冒険の旅を続けていましたが、それはあくまで自分の足で歩ける範囲。不思議の国にもパソコン通信があったなら、時を選ばず、所を選ばず、この奇妙な世界のすべてを体験していたかもしれません。原因と結果がさかさまになったり、何が起こるかわからないこの国こそ、パソコン通信がぴったりの世界。時間と空間を飛び越えるのは、パソコン通信の得意技、ですからね。

パソコン通信なら、こんな楽しさ。

パソコン通信なら、電子メール・BBS・データベース・フリーソフトウェアと、メニューは豊富。万華鏡のようにくるくる変わる「不思議の国」の心ときめかせる楽しさ以上に、魅力的な世界に入り込めます。あなたも、パソコン通信のネットワークの中に飛び込んでみませんか？

きっと、出会える。キットで、会える。

買ったその日から
2週間無料で
アクセスできます。

J&P HOTLINEへのご入会はスタータキットで。お求めは、下記のお店でどうぞ。または現金書留にて、¥3,000+¥90（消費税3%）=¥3,090を、事務局までお送り下さい。すぐにスタータキットをお送りします。

お問い合わせは――――
〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOTLINE事務局宛 TEL(06) 632-2512

スタータキットのお求めはJ&P各店でどうぞ。

渋谷店	東京都渋谷区道玄坂2－28－4	☎(03)3496-4141
町田店	東京都町田市森野1－39－16	☎(0427)23-1313
八王子店	東京都八王子市旭町1-1八王子そごう7F	☎(0426)26-4141
立川店	東京都立川市幸町4－39－1	☎(0425)36-4141
三鷹店	東京都三鷹市野崎1－20－17	☎(0422)31-6251
横浜店	横浜市西区北幸2-9-5横浜HSビル1F	☎(045)313-6711
焼津インター店	静岡県焼津市越後島385	☎(054),626-3311
富山店	富山市掛尾町300	☎(0764)22-5033
金沢店	金沢市入江2－63	☎(0762)91-1130
寺地店	金沢市寺地2－3	☎(0762)47-2524
大須店	名古屋市中区大須4－2－48	☎(052)262-1141
本厚木店	神奈川県厚木市中町3－4－4	☎(0462)25-5151
テクノランド	大阪市浪速区日本橋5－6－7	☎(06) 634-1211
メディアランド	大阪市浪速区日本橋5－8－26	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2－1－17	☎(06) 634-3111
U. S. LAND	大阪市浪速区日本橋4－9－15	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
高槻店	高槻市高槻町11－16	☎(0726)85-1212
くずは店	枚方市楠葉花園町15－2	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3 SENCHU PAL 2番街4F	☎(06) 834-4141
摂津富田店	高槻市大畑町24－10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4－20	☎(0720)34-1166
枚方バイパス店	枚方市田口3－41－7	☎(0720)48-1211
藤井寺店	藤井寺市岡2－1－33	☎(0729)38-2111
岸和田店	岸和田市土生町2451－3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3－2－16	☎(078)231-2111
西宮店	西宮市河原町5－11	☎(0798)71-1171
伊丹店	伊丹市昆陽池1－63	☎(0727)77-5101
姫路店	姫路市東延末1-1住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵比須之町549	☎(075)341-4411
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4－4	☎(0734)28-1441
和歌山南店	和歌山市中島368	☎(0734)25-1414
学園前店	奈良市学園北1－8－10	☎(0742)49-1411
奈良1ばん館	奈良市三条町478－1	☎(0742)27-1111
新大宮店	奈良市法華寺町83－5	☎(0742)35-2611
郡山インター店	大和郡山市横田693－1	☎(07435)9-2221
田原本店	奈良県磯城郡田原本町千代574-1	☎(07443)3-4041
熊本店	熊本市手取本町4－12	☎(096)359-7800

SHARP

目の付けどころが、
シャープでしょ。

いわば“感性”専用。

ことマインドに関しては
「汎用」という概念は存在しないも同じです。
「実用的である」のと、これなら「使える」というのも違います。
X68000が、普通のパソコンとは違うといわれる所以もここにあります。
いわゆる実用性を重視したビジネスパソコンとは
創造力で一線を画しています。
何に使うのか、何がしたいのか、
パソコン選びのポイントは目的にあったマシンを探すこと。
普通のパソコンに合わせるのでは
あなたのせっかくの創造力も発揮されません。
X68000は、使う人のクリエイティブマインドを咲かせる
“感性”専用パソコンです。

X68000
PERSONAL WORKSTATION·XVI
Compact

本体＋キーボード＋マウス
2HD3.5インチFDDタイプ CZ-674C-H（グレー） 標準価格298,000円（税別）
14型カラーディスプレイ（ドットピッチ0.28mm）
CZ-608D-H（グレー） 標準価格94,800円（税別）

- 5.25インチ増設用フロッピーディスクドライブ CZ-6FD5 標準価格99,800円・税別〔接続ケーブル同梱〕
- ディスプレイテレビ/CZ-6TU用RGBケーブル CZ-6CR1 標準価格4,500円・税別
- ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル CZ-6CT1 標準価格5,500円・税別
- SCSI変換ケーブル CZ-6CS1 標準価格12,000円・税別

（カラー液晶ディスプレイとの組み合わせ例）
10.4型TFTカラー液晶ディスプレイ
LC-10C1-H（グレー）標準価格 598,000円（税別）
接続ケーブル AN-1515X 標準価格 4,200円（税別）

※カラー液晶ディスプレイを接続してご使用の場合、
SX-WINDOW上のアプリケーション利用に
限定されます。

●お問い合わせは…
シャープ株式会社 電子機器事業本部システム機器営業部〒545大阪市阿倍野区長池町22番22号☎(06)621-1221（大代表） 電子機器事業本部AVCシステム事業推進室〒162東京都新宿区市谷八幡町8番地☎(03)3260-1161（大代表）

T1002179110602 雑誌 02179-11